Rafael Álvarez · Joan Josep Climent · Francisco Ferrández · Francisco M. Martínez
Leandro Tortosa · José Francisco Vicent · Antonio Zamora
(editores)

Actas de la XIII Reunión Española sobre Criptología y Seguridad de la Información

RECSI XIII

Alicante, 2-5 de septiembre de 2014

PUBLICACIONES UNIVERSIDAD DE ALICANTE

Actas de la XIII Reunión Española sobre Criptología y Seguridad de la Información

RECSI XIII

Alicante, 2-5 de septiembre de 2014

Rafael Álvarez · Joan Josep Climent · Francisco Ferrández · Francisco M. Martínez
Leandro Tortosa · José Francisco Vicent · Antonio Zamora
(editores)

Publicaciones de la Universidad de Alicante

Campus de San Vicente, s/n
03690 San Vicente del Raspeig
Publicaciones@ua.es - http://publicaciones.ua.es
Teléfono: 965 903 480

ISBN: 978-84-9717-323-0

Universitat d'Alacant
Universidad de Alicante

RECSI XIII
ALICANTE 2014

Organización XIII RECSI

Comité Científico

- Abascal Fuentes, Policarpo *(Universidad de Oviedo)*
- Álvarez Sánchez, Rafael *(Universidad de Alicante)*
- Amigó García, José María *(Universidad Miguel Hernández de Elche)*
- Areitio Bertolín, Javier *(Universidad de Deusto)*
- Arenaza Nuño, Ignacio *(Mondragón Unibertsitatea)*
- Borrell Viader, Joan *(Universidad Autónoma de Barcelona)*
- Bras Amorós, Maria *(Universidad Rovira i Virgili)*
- Caballero Gil, Pino *(Universidad de La Laguna)*
- Castellà Roca, Jordi *(Universidad Rovira i Virgili)*
- Climent Coloma, Joan Josep *(Universidad de Alicante)*
- Domingo Ferrer, Josep *(Universidad Rovira i Virgili)*
- Durán Díaz, Raúl *(Universidad de Alcalá)*
- Fernández-Medina Patón, Eduardo *(Universidad de Castilla-La Mancha)*
- Ferrer Gomila, Josep Lluís *(Universidad de las Illes Balears)*
- Fúster Sabater, Amparo *(C.S.I.C.)*
- García Bringas, Pablo *(Universidad de Deusto)*
- García Teodoro, Pedro *(Universidad de Granada)*
- González Vasco, Mª Isabel *(Universidad Rey Juan Carlos)*
- Gutiérrez Gutiérrez, Jaime *(Universidad de Cantabria)*
- Hernández Encinas, Luis *(C.S.I.C.)*
- Hernández Goya, Candelaria *(Universidad de La Laguna)*
- Herrera Joancomartí, Jordi *(Universidad Autónoma de Barcelona)*
- Huguet Rotger, Llorenç *(Universidad de las Illes Balears)*
- Jacob Taquet, Eduardo *(Universidad del País Vasco/Euskal Herriko Unibertsitatea)*
- López Muñoz, Javier *(Universidad de Málaga)*
- Martín del Rey, Ángel *(Universidad de Salamanca)*

- Martínez López,Consuelo *(Universidad de Oviedo)*
- Megías Jiménez, David *(Universitat Oberta de Catalunya)*
- Miret Biosca, José María *(Universidad de Lleida)*
- Morillo Bosch, Paz *(Universidad Politécnica de Catalunya)*
- Muñoz Muñoz, Alfonso *(Universidad Politécnica de Madrid)*
- Peinado Domínguez, Alberto *(Universidad de Málaga)*
- Ramió Aguirre, Jorge *(Universidad Politécnica de Madrid)*
- Ramos Álvarez, Benjamín *(Universidad Carlos III de Madrid)*
- Ribagorda Garnacho, Arturo *(Universidad Carlos III de Madrid)*
- Rifá Coma, Josep *(Universidad Autónoma de Barcelona)*
- Sáez Moreno, Germán *(Universidad Politécnica de Catalunya)*
- Salazar Riaño, José Luis *(Universidad de Zaragoza)*
- Sánchez Ávila, Carmen *(Universidad Politécnica de Madrid)*
- Sebé Feixa, Francesc *(Universidad de Lleida)*
- Soriano Ibáñez, Miguel *(Universidad Politécnica de Catalunya)*
- Tortosa Grau, Leandro *(Universidad de Alicante)*
- Uribeetxeberria Ezpeleta, Roberto *(Mondragón Unibertsitatea)*
- Tena Ayuso, Juan *(Universidad de Valladolid)*
- Vicent Francés, José Francisco *(Universidad de Alicante)*
- Villar Santos, Jorge *(Universidad Politécnica de Catalunya)*
- Zamora Gómez, Antonio *(Universidad de Alicante)*
- Zurutuza, Urko *(Mondragón Unibertsitatea)*

Comité Organizador

- Álvarez Sánchez, Rafael *(Universidad de Alicante, Vicepresidente)*
- Climent Coloma, Joan Josep *(Universidad de Alicante)*
- Ferrández Agulló, Francisco *(Universidad de Alicante)*
- Martínez Pérez, Francisco Miguel *(Universidad de Alicante)*
- Tortosa Grau, Leandro *(Universidad de Alicante)*
- Vicent Francés, José Francisco *(Universidad de Alicante, Vicepresidente)*
- Zamora Gómez, Antonio *(Universidad de Alicante, Presidente)*

Contenidos

Prefacio

Si tuviéramos que elegir un conjunto de palabras clave para definir la sociedad actual, sin duda el término información sería uno de los más representativos. Vivimos en un mundo caracterizado por un continuo flujo de información en el que las Tecnologías de la Información y Comunicación (TIC) y las Redes Sociales desempeñan un papel relevante. En la Sociedad de la Información se generan gran variedad de datos en formato digital, siendo la protección de los mismos frente a accesos y usos no autorizados el objetivo principal de lo que conocemos como Seguridad de la Información.

Si bien la Criptología es una herramienta tecnológica básica, dedicada al desarrollo y análisis de sistemas y protocolos que garanticen la seguridad de los datos, el espectro de tecnologías que intervienen en la protección de la información es amplio y abarca diferentes disciplinas. Una de las características de esta ciencia es su rápida y constante evolución, motivada en parte por los continuos avances que se producen en el terreno de la computación, especialmente en las últimas décadas. Sistemas, protocolos y herramientas en general considerados seguros en la actualidad dejarán de serlo en un futuro más o menos cercano, lo que hace imprescindible el desarrollo de nuevas herramientas que garanticen, de forma eficiente, los necesarios niveles de seguridad.

La Reunión Española sobre Criptología y Seguridad de la Información (RECSI) es el congreso científico español de referencia en el ámbito de la Criptología y la Seguridad en las TIC, en el que se dan cita periódicamente los principales investigadores españoles y de otras nacionalidades en esta disciplina, con el fin de compartir los resultados más recientes de su investigación. Del 2 al 5 de septiembre de 2014 se celebrará la decimotercera edición en la ciudad de Alicante, organizada por el grupo de Criptología y Seguridad Computacional de la Universidad de Alicante. Las anteriores ediciones tuvieron lugar en Palma de Mallorca (1991), Madrid (1992), Barcelona (1994), Valladolid (1996), Torremolinos (1998), Santa Cruz de Tenerife (2000), Oviedo (2002), Leganés (2004), Barcelona (2006), Salamanca (2008), Tarragona (2010) y San Sebastián (2012).

Ponencias invitadas

25 Años de Criptografía con Curvas Elípticas

Juan G. Tena
IMUVA
Universidad de Valladolid
Email: tena@agt.uva.es

***Resumen*—Se describe brevemente el nacimiento y los principales hitos en el desarrollo histórico de la Criptografía con Curvas Elípticas, sus fortalezas y vulnerabilidades. Se examinan las condiciones exigibles a una curva elíptica para ser *criptográficamente fuerte* y las estrategias para encontrar tales curvas. Finalmente se analiza el caso particular de la Criptografía con Curvas Elípticas en el contexto de las tarjetas inteligentes.**

***Palabras clave*—curvas elípticas; logaritmo discreto; curvas criptográficamente buenas; isogenias; pairings; tarjetas inteligentes**

I. Introducción

Las curvas elípticas han ocupado un papel central en Matemáticas desde hace tres siglos y sus notables propiedades, aritméticas y geométricas, han encontrado aplicación en múltiples problemas y campos matemáticos.

Su empleo en Criptografía es sin embargo reciente, pudiéndose situar su inicio en los dos artículos siguientes:

- V. Miller: Use of elliptic curves in Cryptography, CRYPTO'85, 1985, [13].
- N. Koblitz: Elliptic Curve Cryptography, Math. Comp., 1987, [9].

En ambos, los autores proponen implementar el Problema del Logaritmo Discreto (PLD) en el grupo de puntos de una curva elíptica definida sobre un cuerpo finito, en lugar de en el grupo multiplicativo de un tal cuerpo, como se hacía clásicamente. La motivación aducida es que tal grupo de puntos resulta inmune a ataques criptoanalíticos, como el Index-Calculus, lo que permite una seguridad equivalente con longitudes de clave mucho menores.

Sin embargo, la idea de Miller y Koblitz permaneció inicialmente en el ámbito académico y aún en 1997 R. Rivest escribía:

The security of cryptosystems based on elliptic curves is not well understood, due in large part to the abstruse nature of elliptic curves.

La implantación del nuevo paradigma debe mucho a la compañía Certicom (creada en 1985 por S.A. Vanstone y R. Mullin) y al grupo investigador de la Universidad de Waterloo (A.J. Menezes, S.A.Vanstone, etc).

It was enterely Scott Vanstone and his students and collaborators who transformed ECC from a gleam in two mathematicians'eyes to something that was ready from prime time. (N. Koblitz).

Actualmente la Criptografía con Curvas Elípticas (ECC) es una disciplina madura y consolidada, teórica y tecnológicamente. Los libros y artículos, los congresos y seminarios sobre el tema son muy numerosos y compañías e instituciones como NIST, Certicom, IEEE, RSA Laboratories, etc incluyen Criptografía Elíptica en sus standards.

Sin embargo, el camino recorrido no ha estado exento de obstáculos. Algunos de ellos han sido debidos a problemas de implementación:

1. Cálculo del cardinal del grupo de puntos de la curva elíptica. Los algoritmos para su determinación (SEA, T. Satoh, etc, [1], [2]) son costosos.
2. Identificación de los mensajes a cifrar m con puntos P_m de la curva elíptica utilizada.
3. Optimización de las operaciones (suma y multiplicación escalar) con puntos de la curva. Diversos algoritmos, utilizando diferentes tipos de coordenadas (afines, proyectivas, etc) y diferentes ecuaciones para la curva (Weiertrass, Hess, Montgomery, Edwards, etc), [8], [1] han sido propuestos.

Por otra parte, en Criptografía, ninguna propuesta está exenta de vulnerabilidades. En el caso de las curvas elípticas, su rica estructura matemática es un arma de doble filo, ya que puede ser explotada también por el criptoanálisis.

Un ejemplo clásico es el algoritmo de Menezes–Okamoto–Vanstone (MOV, 1993), [14], que permite reducir el PLD para una curva elíptica E, definida sobre el cuerpo finito $\mathbf{F}_q$, al PLD sobre un cuerpo extensión $\mathbf{F}_{q^k}$, para un cierto número k (dependiente de E) al cual se denomina *grado de inmersión de la curva* E. El algoritmo MOV utiliza como herramienta el denominado pairing de Weil. Los pairing (de Weil, Tate, etc) son aplicaciones bilineales definidas sobre una curva elíptica y con valores en un grupo cíclico, ver [2], [6].

La utilidad criptoanalítica del algoritmo MOV depende del grado de inmersión k, ya que solo resulta eficiente si k es pequeño. En particular Menezes, Okamoto y Vanstone muestran que esto ocurre para las curvas elípticas denominadas supersingulares, en las que el valor de k es a lo sumo 6. Por ello tales curvas se consideran vulnerables para criptosistemas basados en el PLD. Cabe señalar que, sin embargo, las curvas

supersingulares son idóneas para su empleo en otra rama de la Criptografía, la Criptografía Basada en la Identidad, [12].

La idea de Criptografía Basada en la Identidad fue introducida por A. Shamir (CRYPTO 1984), ver [12]. En ella la clave pública de un usuario puede deducirse de su nombre (o cualquier otra información relacionada con su identidad). Shamir propuso esquemas de firma e intercambio de claves basadas en la identidad, pero no sistemas de cifrado. D. Bonech y M. Franklin (CRYPTO 2001), [3], proponen un criptosistema basado en la identidad, utilizando el pairing de Weil sobre curvas elípticas.

Actualmente existen propuestas de criptosistemas, esquemas de firma y esquemas de intercambio de claves basadas en pairings. A diferencia de los criptosistemas basados en el PLD elíptico la Criptografía basada en la identidad requiere curvas elípticas con grado de inmersión pequeño, las denominadas *pairing friendly curves*.

Otra herramienta, propia de las curvas elípticas, que permite el diseño de criptosistemas y protocolos criptográficos son las isogenias. Una isogenia es una aplicación lineal entre dos curvas elípticas, ver [6]. En el caso de un cuerpo base finito $\mathbf{F}_q$ dos curvas elípticas, definidas sobre $\mathbf{F}_q$, son isógenas (es decir existe una isogenia no nula entre ellas) si y solo si tienen igual cardinal.

Sin embargo, dadas dos curvas elípticas sobre $\mathbf{F}_q$ y con igual cardinal, encontrar explícitamente una isogenia entre ambas es un problema computacionalmente difícil, lo que posibilita utilizar este problema en el diseño de criptosistemas de clave pública. A. Rostovsov y A. Stolbunov (Eurocrypt' 2006), proponen un criptosistema basado en *estrellas* de isogenias, [18].

Como un último ejemplo de la fecundidad de las curvas elípticas citemos su aplicación en dos técnicas auxiliares del criptosistema RSA, métodos de factorización y tests de primalidad:

- Métodos de factorización: algoritmo de H.W. Lenstra Jr., 1987, [1], [6].
- Tests de primalidad: test de S. Goldwaser–J. Kilian, 1996 y test de A.O.L. Atkins–F. Morain, 1993, [4].

II. Curvas Elípticas

En la sección anterior hemos resumido la aparición y el desarrollo de la Criptografía basada en curvas elípticas, pero no hemos precisado que son tales curvas. Podemos tomar como definición la siguiente, [1], [6],

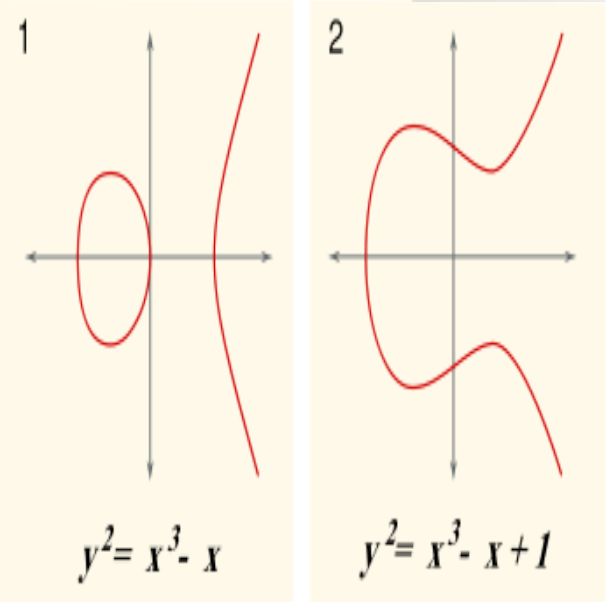
Definición 1: Una *Curva Elíptica* E definida sobre un cuerpo k (por ejemplo el cuerpo de los números complejos $\mathbf{C}$, los reales $\mathbf{R}$, un cuerpo finito $\mathbf{F}_q$, etc) es una curva proyectiva, no singular, admitiendo una ecuación, definida sobre k, en la denominada *Forma Normal de Weierstrass*:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6,\ a_i \in k$$

Una tal curva admite un único punto en el infinito, el $O = (0 : 1 : 0)$ (punto del infinito en la dirección del eje **y**). Si $Car(k) \neq 2, 3$ (es decir cuerpo no binario ni ternario) la forma de Weiertrass puede reducirse a la ecuación más simple:

$$E : y^2 = x^3 + Ax + B,\ A, B \in k$$

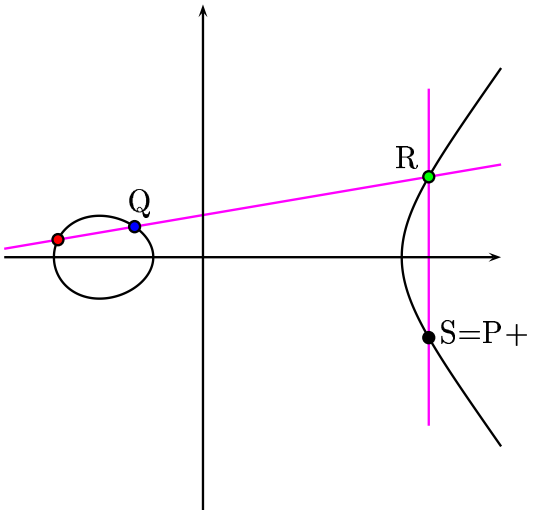
La gráfica de esta curva, para el cuerpo $\mathbf{R}$ de los números reales, toma una de las dos formas siguientes:



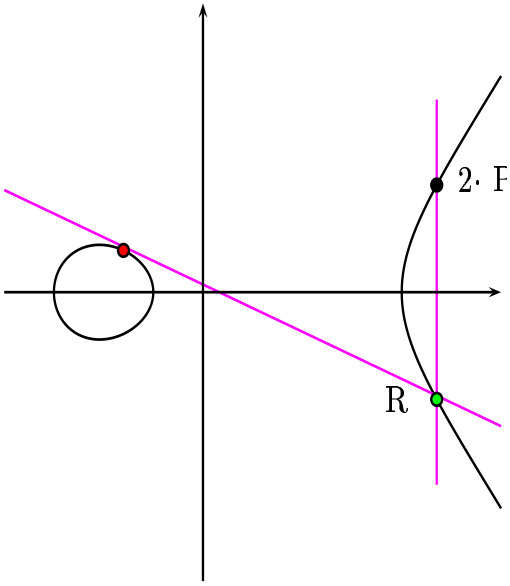
Denotaremos con $E(k)$ al conjunto de puntos de la curva E con coordenadas en el cuerpo k (incluido el punto en el infinito O). La utilidad de las curvas elípticas deriva de la posibilidad de dotar a $E(k)$ de una estructura de grupo abeliano (con O como elemento neutro). La ley de grupo puede definirse geométricamente. Por simplicidad supongamos Car$(k) \neq 2, 3$ y $E : y^2 = x^3 + Ax + B$ y recordemos que, por el teorema de Bezout, una recta L corta a E en tres puntos.

Definición 2: La suma de dos puntos $P, Q \in E(k)$ es el punto simétrico, respecto del eje x, del tercer punto de intersección con la cúbica de la recta que une P y Q. Si $P = Q$, (en cuyo caso se habla de doblado del punto) se sustituye cuerda por tangente.

Las dos figuras siguientes muestran gráficamente las operaciones de suma y doblado de puntos:



Suma de Puntos en Curva Elíptica



Doblado de Punto en Curva Elíptica

Consideremos ahora el caso particular de un cuerpo base finito $k = \mathbf{F}_q,\ q = p^m$. Se tiene entonces, [1], [6].

Teorema 3: Sea E una curva elíptica definida sobre $\mathbf{F}_q$.

1. Sea $N = \#(E(\mathbf{F}_q))$ el cardinal de la curva. Se verifica el teorema de Hasse:
$$q + 1 - 2\sqrt{q} \leq N \leq q + 1 + 2\sqrt{q}$$
Es decir $N = q + 1 - t,\ \text{con}\ |t| \leq 2\sqrt{q}$.
2. El grupo abeliano finito tiene la estructura siguiente, $E(\mathbf{F}_q) \simeq \mathbf{Z}/n_1\mathbf{Z} \times \mathbf{Z}/n_2\mathbf{Z}$
donde $N = n_1 n_2,\ \ n_2|n_1,\ \ n_2|q-1$.

En la Introducción se ha hecho referencia a las curvas elípticas supersingulares,

Definición 4: Una curva elíptica E definida sobre el cuerpo finito $\mathbf{F}_q,\ q = p^m,$ se llama supersingular si p divide a t

III. Logaritmo Discreto Elíptico

Recordemos el Problema del Logaritmo Discreto (PLD), [6].

Definición 5: Sea $G = \langle g \rangle$ un grupo cíclico finito con cardinal N (clásicamente el grupo G considerado era $\mathbf{F}_q^* = \mathbf{F}_q - \{0\}$ con cardinal $N = q - 1$). Si $x \in G$, se denomina logaritmo discreto de x en la base g al entero natural $n \leq N$ tal que $g^n = x$.

Conocidos g y n es computacionalmente sencillo calcular x. Sin embargo conocidos g y x, es computacionalmente intratable determinar n (Problema del Logaritmo Discreto).

Diversos sistemas criptograficos (J. Massey–J. Omura, T. ElGamal), esquemas de firma electronica (T. ElGamal, C. P. Schnorr, DSA) e intercambio de claves (W. Diffie–M. E. Hellman) están basados en el PLD, [8].

La seguridad del logaritmo discreto ha sido exhaustivamente estudiada. Podemos clasificar los algoritmos para resolver el PLD en tres tipos, [6]:

1. Algoritmos válidos en cualquier grupo (todos los cuales tienen un coste exponencial): Rho de J. M. Pollard, Baby Steps Giant Steps (BSGS), etc.
2. Algoritmo de R. Silver– G. C. Pohlig–M.E. Hellman: Eficiente para grupos cuyo cardinal tiene todos sus factores primos *pequeños*. En consecuencia el cardinal del grupo debería poseer un factor primo grande para ser seguro.
3. Algoritmos tipo *Index Calculus*.

El método del Index-Calculus se ha aplicado con éxito (coste subexponencial) a los cuerpos finitos $\mathbf{F}_q$, en particular los binarios $\mathbf{F}_{2^m}$. Actualmente se considera necesario un tamaño mínimo para el cardinal de estos cuerpos de 1024 bits, lo que obliga a aumentar el tamaño de las claves y por tanto los recursos computacionales necesarios.

Una posibilidad alternativa es substituir el grupo $G = \mathbf{F}_q^*$ por otros inmunes al Index Calculus. Esta fué la motivación de Miller y Koblitz para su propuesta del Problema del Logaritmo Discreto Elíptico (PLDE): Dada una curva elíptica E sobre $\mathbf{F}_q$ y puntos P y $Q = nP$ en $E(\mathbf{F}_q)$ encontrar n.

El PLDE ofrece las siguientes ventajas sobre el PLD clásico:

- Flexibilidad: Fijado el cuerpo $\mathbf{F}_q$ existen muchas curvas elípticas sobre él, lo que ofrece la posibilidad de cambiar periódicamente la curva, manteniendo $\mathbf{F}_q$ (y su aritmética).
- El grupo $E(\mathbf{F}_q)$ es inmune al Index Calculus, lo que lo hace más seguro que el grupo $\mathbf{F}_q^*$:
  1. El ataque al PLDE (utilizando el algoritmo de Pollard) para una curva elíptica sobre $\mathbf{F}_p$, p primo de 160 bits, exige aproximadamente 10^{24} operaciones elementales.
  2. El ataque al DLP (utilizando el método del Index Calculus) para $\mathbf{F}_p^*$, p primo de 160 bits, necesita solo 10^9 operaciones.

- Esta posibilidad de claves más cortas hace especialmente idónea a la Criptografía con curvas elípticas para su uso en plataformas con capacidad computacional reducida como tarjetas inteligentes, RFID, redes de sensores, etc.

Comparación de tamaños de clave (NIST)

PLD/RSA	$PLDE$	$Ratio$
1024	163	1 : 6
3072	256	1 : 12
7680	384	1 : 20
15360	512	1 : 30

Ataques al PLDE, como el ya mencionado método MOV, propiciaron la búsqueda de otras alternativas como base del logaritmo discreto. Es el caso de las Curvas Hiperelípticas, [10], generalización de las elípticas. Estas curvas vienen dadas por una ecuación del tipo:

$$C : y^2 + h(x)y = f(x) \mid \ gr(h) \leq g, \ gr(f) = 2g + 1$$

(Las curvas elipticas corresponden al caso $g = 1$).

Sin embargo el PLD sobre (las jacobianas de) curvas hiperelípticas se ha mostrado vulnerable, para $g > 2$, frente a variantes del Index Calculus: algoritmos de L. M. Adleman–J. De Marrais–M. D. Huang (1994) y de P. Gaudry (2000), [2], [6].

III-A. Curvas elípticas criptográficamente buenas

Aunque actualmente el PLDE se considera seguro, algunas precauciones son necesarias en la elección de la curva elíptica base E:

1. El cardinal de E debe ser adecuado (primo o con un factor primo grande) para evitar el ataque de Silver-Pohlig-Hellman.
2. Con grado de inmersión "grande"(en particular no supersingulares) para evitar el ataque MOV.
3. Evitar las denominadas curvas *Anómalas*, curvas sobre $\mathbf{F}_p$ (p primo), y con cardinal p, curvas para las que el PLDE es fácil: ataques de I.A. Semaev (1998), T. Satoh–K. Araki (1998) y N. Smart (1999), [6].
4. E debe ser inmune al ataque por Descenso de Weil, tipo de ataque propuesto por G. Frey en 2001 y desarrollado por P. Gaudry, A.J. Menezes, N. Smart, etc [2].

Dos vías, ambas costosas, pueden utilizarse para elegir una curva elíptica *buena* (a salvo de las debilidades mencionadas):

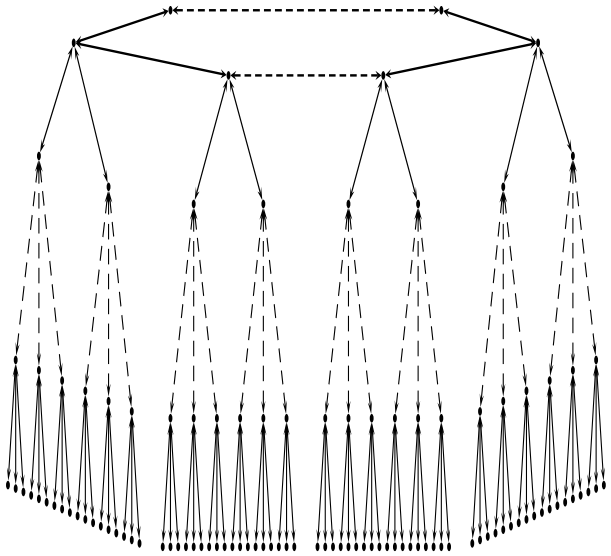
- Tomar curvas aleatoriamente, calcular su cardinal y comprobar si es adecuado.
- Construcción de curva elíptica con cardinal adecuado prefijado. Ello es factible empleando un método debido a A. O. L. Atkin–F. Morain, [4]

Una tercera vía consiste en el empleo de isogenias. Como se ha señalado, dos curvas isógenas tienen igual cardinal. Por tanto, partiendo de una curva *criptográficamente buena* (con cardinal adecuado N), todas las curvas obtenidas a partir de ellas como imágenes por isogenias serán también buenas. Ello justifica el estudio de tales relaciones de isogenia.

Consideremos el conjunto de todas las curvas elípticas (definidas salvo isomorfía) sobre un cuerpo finito dado $\mathbf{F}_q$, $q = p^m$ y con cardinal N. Sea ℓ un primo diferente de la característica p del cuerpo y consideremos todas las posibles isogenias de grado ℓ entre tales curvas (ver [6] para el concepto de grado de una isogenia). Tal conjunto puede considerarse como un grafo dirigido $\mathcal{G}(\mathcal{N}, \ell)$, con aristas dichas ℓ-isogenias. Es posible asignar a estas aristas un cierto sentido (horizontal, ascendente o descendente) y por tanto estratificar a $\mathcal{G}(\mathcal{N}, \ell)$ en pisos o niveles, [11].

Definición 6: Cada componente conexa de $\mathcal{G}(\mathcal{N}, \ell)$ se denomina un ℓ-volcán.

El nombre de volcán responde a su similitud con un cono volcanico, de hecho en un ℓ-volcán se habla de cráter, ladera y suelo. La noción de grafo de ℓ-isogenias y ℓ-volcanes se debe a D. R. Kohel (Ph. D. Thesis, 1996), [11] y posteriormente su estructura y propiedades han sido estudiadas por otros investigadores: M. Fouquet–F. Morain (LNCS 2369, 2002) [5], J.Miret–R.Moreno–D.Sadornil–J.Tena–M.Valls (Applied Mathematics and Computation, 2006 y 2008), [15], etc.



Estructura de un 3-Volcán

El grafo total $\mathcal{G}(\mathcal{N}, \ell)$ está formado por varios ℓ-volcanes y puede denominarse una ℓ-cordillera, [16].

IV. Curvas Elípticas y Tarjetas Inteligentes

Aunque las primeras tarjetas de crédito se remontan a 1950 (Diners Club), las Tarjetas Inteligentes (Smart Cards), con chip incorporado, se popularizan a partir de 1980 (en 1986 se define el standard ISO para ellas) y se integran en la telefonía móvil (Tarjetas SIM: Subscriber Identificatio Module) a partir de 1990.

Los usos actuales de las tarjetas hacen necesaria la implantación en las mismas de sistemas criptográficos (esquemas de cifrado, certificados, firma digital, etc).

La posibilidad ya mencionada de claves más cortas y la flexibilidad en la elección de las mismas, convierten a las curvas elípticas en candidatos privilegiados para la Criptografía implementada en tarjetas inteligentes:

- En 1996 un grupo de empresas, Europay, MasterCard y VISA (EMV) definieron la especificación de tarjetas inteligentes para su uso en servicios financieros.
- En 2001 este grupo propone el empleo (EMV 40 Elliptic Curve Technical Report), de curvas elípticas como alternativa al RSA para Autenticación Estática de Datos (SDA), Autenticación Dinámica de Datos (DDA) y cifrado autónomo.

Además de los ataques criptoanalíticos específicos, propios del sistema criptográfico concreto utilizado, la Criptografía en tarjetas es susceptible de un tipo de ataques activos denominados *Side Channel Attacks*, [2]. Estos ataques se basan en que la alimentación y el reloj de las tarjetas inteligentes son proporcionados por el lector.

Es posible entonces, si se tiene acceso a la tarjeta y se dispone de instrumentos adecuados, medir el consumo, tiempo de computación, etc de la tarjeta, mientras ésta realiza operaciones criptográficas. Tal información puede ser usada por un atacante para obtener la clave privada guardada en la tarjeta. Veamos en particular un tipo de side channel attacks específico de la Criptografía con curvas elípticas.

IV-A. Zero-Value Point Attacks

L. Goubin, 2003, [7], muestra como un atacante puede detectar, midiendo el consumo de la tarjeta, la aparición de puntos de la curva con abscisa u ordenada nulos y después de varias ejecuciones, conseguir la clave secreta d almacenada en la tarjeta. De forma más precisa, suponiendo que un cierto bit de la clave d es 0 ó 1, el atacante intenta crear un punto con alguna de sus coordenadas cero. Si tal punto realmente aparece su suposición era correcta. Posteriormente T. Akishita y T. Takagi (LNCS 2851, Springer, 2003) generalizan el ataque de Goubin a curvas en las que algunos parámetros intermedios usados en el doblado y suma de puntos de la curva elíptica son cero.

¿Es posible obtener curvas inmunes a los ZVPA? N. Smart (LNCS 2779, Springer, 2003) y Akishita-Takagi proponen, partiendo de una curva *buena* (con cardinal adecuado, etc) buscar curvas isogenas a la dada hasta encontrar una adecuada (en particular sin puntos con coordenadas nulas). Para ello construyen ℓ-isogenias, para sucesivos primos ℓ, hasta encontrar una curva resistente.

Un problema de este método es que el coste de las ℓ -isogenias aumenta con el grado ℓ, como puede apreciarse en la tabla adjunta para la curva 192r1 del SECG (Standard for Efficient Cryptography):

ℓ	tiempo (seg.)
5	0.04
11	0.91
13	5.97
23	44.30
37	267.04
59	995.20
73	3474.73

Si una curva resistente no se encuentra para los primeros primos ℓ, el coste de encontrar una curva buena puede ser disuasorio.

Un método alternativo (J.M. Miret–D. Sadornil–J. Tena–R. Tomas–M. Valls, [17], consiste en la construcción de caminos de isogenias mediante búsquedas en los ℓ volcanes. Tal método encuentra la forma más rápida para ir de una curva vulnerable dada a otra resistente.

La tabla siguiente compara los resultados obtenidos (para la curva 192r1 del SECG) con el método de Smart y el alternativo. Con el método de Smart la primera curva resistente se obtiene mediante una 23-isogenia. En nuestro caso se obtiene con una 5-isogenia seguida de una 13-isogenia.

192r1	Smart	Nuestra propuesta
Grado Isógena resistente	23	5-13
Tiempo de cálculo (seg.)	44.30	6.01
Tiempo de la búsqueda (seg.)	51.24	6.99

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto de investigación MTM2010-21580-C02-02 del Ministerio de Economía y Competitividad.

REFERENCIAS

[1] I. Blake, G. Seroussi, N. Smart, “Elliptic Curves in Cryptography”, London Mathematical Society Lecture Note Series 265, Cambridge University Press, 2000.
[2] I. Blake, G. Seroussi, N. Smart, “Advances in Elliptic Curves in Cryptography”, London Mathematical Society Lecture Note Series 317, Cambridge University Press, 2005.
[3] D. Bonech, M. Franklin, “Identity-based encryption from the Weil pairing”, CRYPTO 2001, LNCS 2139, pp. 213-229, Springer, 2001.
[4] R. Crandall, C. Pomerance, “Prime Numbers”, Second Edition, Springer, 2005.
[5] M. Fouquet, F. Morain, “Isogeny Volcanoes and the SEA Algorithm”, Proc. ANTS-V, LNCS 2369, pp. 276-291, Springer, 2002.
[6] S.D. Galbraith, “The Mathematics of Public Key Cryptography”, Cambridge U. Press, 2012.
[7] L. Goubin, ““A refined power-analysis attack on elliptic curve cryptosystems”, PKC 2003, LNCS 2567, pp. 199-211, Springer, 2003.
[8] D. Hankerson, A. Menezes, S. Vanstone, “Guide to Elliptic Curve Cryptography”, Springer, 2004.
[9] N. Koblitz, “Elliptic curve cryptosystems”, Math. Comp. 48, pp. 203-209, 1987.
[10] N. Koblitz, “Hyperelliptic criptosystems”, J. Crypt. 1, pp. 139-150, 1989.
[11] D. R. Kohel, “Endomorphism rings of elliptic curves over finite fields”, Ph. D. Thesis, U. California, Berkeley, 1996.
[12] L. Martin, “Identity-Based Encryption”, Artech House, 2008.

[13] V.S. Miller, “Use of elliptic curves in Crypography”, CRYPTO 1985, LNCS 218, pp. 417-426, Springer, 1986.
[14] A. Menezes, “Elliptic Public Key Cryptography”, Kluwer, 1993.
[15] J. Miret, R. Moreno, D. Sadornil, J. Tena, M. Valls, “Computing the height of volcanoes of ℓ–isogenies of elliptic curves over finite fields”, Applied Mathematics and Computation 196, no. 1, pp. 67-76, 2008.
[16] J. Miret, D. Sadornil, J. Tena, R. Tomas, M. Valls, “Exploiting Isogeny Cordillera Structure to Obtain Cryptography Good Elliptic Curves”, J. Research and Practice in Information Technology, Vol. 47, no 4, pp. 255-265, 2008.
[17] J. Miret, D. Sadornil, J. Tena, R. Tomas, M. Valls, “On avoiding ZVP attacks using isogeny volcanoes”, LNCS 5379, pp. 266-277, Springer, 2009.
[18] A. Rostovtsev, A. Stolbunov, “Public-key cryptosystem based on isogenies”, Cryptology ePrint Archive, Report 2006/145, 2006, http://eprint.iacr.org/.

Juan G. Tena

Juan Tena Ayuso es doctor en Teoría de Números por la Universidad de Grenoble (1973) y en Matemáticas por la Universidad Complutense de Madrid (1973). Durante el periodo 1973-82 fue profesor en las Universidades de Valladolid, Complutense de Madrid y Santander y desde 1983 es catedrático de Algebra en la Universidad de Valladolid.

Sus áreas de investigación incluyen Criptografía, Teoría de Números y Códigos Correctores de Errores y en la actualidad su interés se centra en el estudio de la Aritmética y Geometría de las Curvas Elípticas y sus aplicaciones criptográficas. Sobre estos temas tiene publicados numerosos libros y artículos científicos e impartidos cursos, conferencias y seminarios. Ha dirigido seis tesis doctorales en el campo de la Criptografía y los Códigos Correctores y ha sido investigador principal y miembro investigador de proyectos de investigación subvencionados por los diversos ministerios españoles competentes (Educación, Educación y Ciencia, Ciencia y Tecnología, Ciencia e Innovación, Economía y Competitividad), Junta de Castilla y León, Unión Europea y Acciones Integradas Hispano-Francesas. Ha sido organizador y miembro de los comités científicos de diversos congresos de su especialidad así como árbitro, recensor y evaluador de artículos científicos y proyectos de investigación.

Moti Yung

Moti Yung es especialista en criptografía e informático en Google Inc. Obtuvo su título de Doctor en la Universidad de Columbia en 1988, y trabajó en el Centro de Investigación de IBM Thomas J. Watson. Ha sido vicepresidente y científico en CertCo, además de director de Investigación sobre Autenticación Avanzada en los Laboratorios RSA. También ha ocupado puestos como profesor adjunto y visitante en la Universidad de Columbia, donde ha dirigido a varios estudiantes de doctorado. En una publicación del año 1996, junto con Adam Young, Yung acuñó el término «criptovirología» para el uso de la criptografía como herramienta de ataque usada por virus y otros tipos de malware (prediciendo ataques de tipo ransomware o rescate). Young y Yung son los autores del libro: Malicious Cryptography: Exposing Cryptovirology (John Wiley & Sons, 2004). También introdujeron la noción de «Cleptografía» para describir el uso de sistemas criptográficos dentro, a su vez, de otros sistemas criptográficos como herramientas de ataque para diseñadores y creadores donde la criptología maliciosa incrustada tiene fuertes propiedades de seguridad contra ingeniería inversa (además, este tipo de ataques han sido publicados y supuestamente se han empleado).

El Dr. Yung ha contribuido extensamente a la creación de los protocolos y sistemas criptográficos básicos (como la noción de seguridad con texto cifrado escogido, que actualmente es un requerimiento fundamental de los sitemas de cifrado de clave pública que funcionan en Internet). Su contribución también abarca muchas áreas de investigación en criptografía y seguridad de los datos, así como estructuras conducentes a su utilización práctica e implementaciones en sistemas y redes. El Dr. Yung recibió el premio anual Distinguished Lecturer in Cryptography de la International Association for Cryptologic Research (IACR) en el Eurocrypt 2010. En 2013 se convirtió en miembro de la Association for Computing Machinery (ACM) y en 2014 de la IACR.

Criptología

A new linear consistency test attack on noised irregularly clocked linear feedback shift registers

Slobodan Petrović
NISlab, Department of Computer Science and Media Technology
Gjøvik University College, p.o. box 191, N-2802 Gjøvik, Norway
Email: slobodan.petrovic@hig.no

***Abstract*—Linear Consistency Test (LCT) is a widely used algebraic attack against pseudorandom generator schemes. A system of linear equations depending on a guessed part of the key is assigned to the analyzed generator and checked for consistency. If the guessed part of the key is not the right one, the system will be inconsistent with high probability. In the presence of noise, additional measures are necessary for this attack to be successful. They must reduce the influence of intercepted output bits complemented by noise. In this paper, a technique is described that tries to guess which bit(s) of the intercepted output sequence are complemented by noise and remove all the equations from the linear system assigned to the generator that depend on those bits. The technique is demonstrated on cryptanalysis of a Binary Rate Multiplier (BRM). The experiments on this generator show that such an attack is feasible if the noise level is up to moderate.**

***Index Terms*—Cryptanalysis, Irregular Clocking, Linear Consistency Test (LCT), Linear Feedback Shift Register (LFSR)**

I. Introduction

Linear Consistency test (LCT) is an algebraic attack against a pseudorandom generator scheme that tries to recover its whole initial state starting from some guessed bits of it. A linear system in the unknown bits of the key is set up, in which the right-hand side of every equation is an intercepted bit of the output sequence of the generator, and its consistency is checked. If the guessed part of the key is not the right one, such a system will be inconsistent with high probability, see [7]. The attack proceeds as follows: first, a subset of the key bits is guessed. Then, a system of equations, in which the rest of the key bits are variables is set up. If the guessed key bit subset is not the right one, the consistency probability of the obtained system will be very small. The consistency of the system is tested for all the possible choices of the guessed portion of the key. If the length of the intercepted output sequence of the generator is sufficient (see [7]), the right choice of the guessed part of the key will lead to a consistent system, whose solution will be the rest of the key.

If we consider a ciphertext-only attack scenario, in which some of the intercepted output bits of the generator are degraded (i.e. complemented) by noise, we have to compensate the influence of the complemented intercepted bits on the consistency of the system in order for the attack to be successful. In this paper, we first guess which of the intercepted bits are complemented by noise and then we remove from the system assigned to the generator all the equations involving the guessed complemented bits. Suppose that the length of the intercepted output sequence is sufficient for making decisions about consistency of the system assigned to the analyzed generator. If the choice of the guessed portion of the key is right, and the guess of the positions of the intercepted bits affected by noise is right, the remaining system will be consistent. If the guessed portion of the key is the right one but the choice of the positions of the intercepted output bits degraded by noise is wrong, the resulting system might be consistent (the probability for this is significant). But if the guess of the key portion is wrong, the system will remain inconsistent with high probability even if the guess of the positions of the bits of the intercepted output sequence degraded by noise is right.

We apply the attack described in the previous paragraph on a representative of a special class of pseudorandom sequence generators, so-called *irregular clocking* generators. Specifically, we analyze how the new LCT attack can be applied on a noised Binary Rate Multiplier (BRM) [2], but the same ideas can be applied in attacks against other representatives of the class as well - the Stop/Go generator, the Shrinking generator, the Alternative Step generator and so on.

The attacks against BRM have been studied by many authors, since that scheme is widely used in practice due to the desirable properties of the output sequence achievable with it (extremely long period and linear complexity, good statistical properties etc.) Most attacks against BRM are *correlation attacks* (for example [3], [4], etc.) The LCT attack against BRM has also been attempted [5], but in a known-plaintext attack scenario, i.e. without noise. The possibility of an LCT attack against noised BRM was studied in [1] and [6]. This attack tries to avoid influence of the bits of the intercepted sequence affected by noise by changing the starting point in the intercepted sequence from which the setting up of the system of equations starts. The attack might be successful if the noise level (i.e. the probability of '1' in the noise sequence) is small, but false alarms and missing the event regarding the right guess of the part of the key are inevitable and because of that a more precise localisation of the bits affected by noise is needed.

The structure of this paper is as follows: In Section II, the BRM-based pseudorandom sequence generator is described. In Section III, the detailed description of the new attack is given. In Section IV, the experimental results are presented

and discussed. Section V concludes the paper.

II. The analyzed generator

We analyze the LCT attack on a noised pseudorandom sequence generator involving a primitive known as The Binary Rate Multiplier (BRM). BRM consists of 2 linear feedback shift registers (LFSRs). One of them, the clocking LFSR (LFSR$_s$), determines the clocking sequence for the clocked LFSR (LFSR$_u$), see Fig. 1.

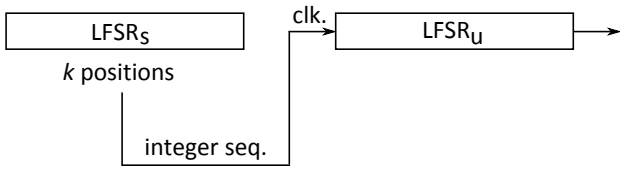


Fig. 1. The BRM primitive

The BRM operates as follows (Fig. 2): Without clocking by LFSR$_s$, the register LFSR$_u$ produces the binary sequence u_n. At the clock pulse i of LFSR$_s$, the bits from k positions of LFSR$_s$ determine the integer s_i that represents the number of bits from the sequence u_n that are going to be discarded. The integers s_i, $i = 1, 2, \ldots$ make the sequence s_n. The process of discarding bits in this way is called *non-uniform decimation* of the sequence u_n. The maximum value of the integer s_i determines the maximum number of bits from the sequence u_n that can be discarded at a time. The binary sequence z_n is the output sequence of the whole BRM.

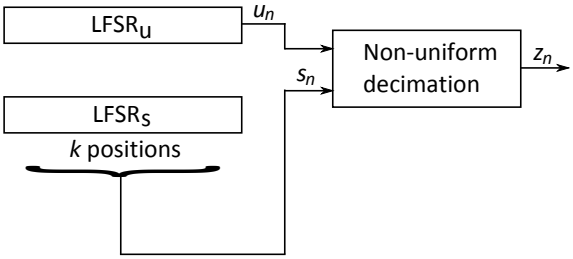


Fig. 2. Operation of the BRM

The BRM primitive has become popular in the design of stream ciphers since it can be shown [2] that the produced sequence z_n has extremely long period and high linear complexity preserving at the same time good statistical properties of a single LFSR.

III. The new LCT attack

In this section, we give details of the new LCT-based attack against a noised BRM. The general description and remarks about LCT have been exposed in the Introduction. To design an LCT attack against BRM, we have to determine which part of the BRM key (which consists of the initial states of LFSR$_s$ and LFSR$_u$, as usual) is to be guessed. It is shown in [5] that assigning a linear system to a BRM when the initial state of LFSR$_s$ is guessed is easy. Then the unknowns in the system are the bits of the output sequence of LFSR$_u$ without decimation together with the bits of the initial state of the same LFSR and the right-hand side of any equation in the system is the corresponding bit of the intercepted sequence. In our new LCT attack on a noised BRM, we use the same approach. We guess the initial state of LFSR$_s$ and make a system of linear equations in the unknowns of the initial state of LFSR$_u$ and the unknown bits of the output sequence of LFSR$_u$ without decimation. The main point of our attack is the algorithm that eliminates the influence of the bits of the intercepted sequence complemented by noise.

Example 1

Suppose the BRM from Fig. 1 uses 4-bit LFSRs and the primitive feedback polynomials of LFSR$_s$ and LFSR$_u$ are $f_s(x) = 1 + x + x^4$ and $f_u(x) = 1 + x^3 + x^4$, respectively. Let the number of output taps of LFSR$_s$ be $k = 2$ and the tap positions are the first and the second (from the left). Let the initial states of LFSR$_s$ and LFSR$_u$ be 1010 and 0110, respectively. Then the clocking sequence for LFSR$_u$ (i.e. the integer sequence s_n) is $31021002333\ldots$ and the output sequence of the BRM is $11010110111\ldots$

Let the cryptanalyst's guess of the initial state of LFSR$_s$ be right, i.e. 1010. In the LCT attack against the generator without noise, the so-called *decimation sequence* is generated, containing the symbol '2' in the positions of the unknown bits. Each symbol '2' will correspond to a new variable in the system of equations assigned to the generator. In this case, the decimation sequence will be $2222 \mid 22212102212011220222122212221\ldots$. The symbol $\mid$ delimits the variables of the initial state of the clocked register LFSR$_u$ from the rest of the variables. The variables to the left from the symbol $\mid$ are given in the order x_4, x_3, x_2, x_1, whereas the variables to the right from the symbol $\mid$ are given in the increased order of indexing, i.e. x_5, x_6, etc. Then the system of linear equations assigned to the given BRM is:

$$\begin{aligned} x_3 + x_4 + x_5 &= 0 \\ x_2 + x_3 + x_6 &= 0 \\ x_1 + x_2 + x_7 &= 0 \\ x_1 + x_5 &= 1 \\ x_5 + x_6 + x_8 &= 0 \\ &\vdots \end{aligned}$$

□

The new ciphertext-only attack against a BRM is described below:

1. Guess the initial state of the LFSR$_s$.
2. Set up a system of equations assigned to such a BRM without involving the intercepted bits. Such a system is homogeneous and always consistent.
3. Set up a system of equations involving only the equations containing the intercepted bits.
4. Join the obtained systems and check the consistency of the joint system. The following cases are possible:

   4.1 There is no noise and the right initial state of LFSR$_s$ was guessed - the joint system will be consistent and

the missing bits of the initial state of LFSR$_u$ can be obtained by solving the system.

4.2 There is no noise and the guess of the initial state of LFSR$_s$ was wrong - the joint system will be inconsistent with high probability, see [7].

4.3 The intercepted sequence was degraded by noise and the guess of the initial state of LFSR$_s$ was right - the joint system will be inconsistent with high probability because of the bits of the intercepted sequence complemented by noise.

4.4 The intercepted sequence was degraded by noise and the guess of the initial state of LFSR$_s$ was wrong - the joint system will be inconsistent with high probability.

5. Suppose that t bits of the intercepted sequence were degraded by noise. Guess their positions. Remove all the equations involving these complemented bits from the joint system. Provided the intercepted sequence is long enough (see [7]), the following cases are possible:

5.1 The guess of the initial state of LFSR$_s$ was right - if the guess of the positions of the bits of the intercepted sequence that were complemented by noise was right, the system will become consistent and solving the system will give the missing bits of the key of the BRM. If the guess of the positions of the bits of the intercepted sequence complemented by noise was wrong, there will be relatively high probability (compared with that in the case of a wrong guess of the initial state of LFSR$_s$) that the system will become consistent.

5.2 The guess of the initial state of LFSR$_s$ was wrong - the system will not become consistent even if the guess of the positions of the bits of the intercepted sequence complemented by noise was right.

6. Repeat the step 5. of the algorithm for all the combinations of guesses for the positions of the t bits complemented by noise in the intercepted sequence, starting from $t = 1$, then $t = 2$ and so on, until a consistent system is obtained.

The success of the attack described above depends on the level of noise, i.e. the ratio between the number of bits complemented by noise in the intercepted sequence and the length of the intercepted sequence. A relatively small level of noise ensures relatively small number of combinations for the guesses of the complemented bits, which makes the attack feasible. In that case the attack is likely to be successful.

Example 2

Refer to the Example 1 above and suppose that the first bit of the intercepted output sequence from the generator is complemented by the noise sequence. This affects the equation $x_1+x_5 = 1$ from Example 1 and other equations involving the complemented bit (2 more equations, since the weight of the feedback polynomial of LFSR$_u$ is 2). The system of linear equations assigned to the given BRM becomes inconsistent (with high probability). To mitigate this, we have to remove the equations from the system that involve the complemented bits (in our example, the number of complemented bits in the intercepted sequence is $t = 1$). We guess the position of the complemented bit. Suppose our guess is right, i.e. the first bit in the intercepted sequence is complemented by noise. If we remove all the equations involving that bit from the system, it will become consistent again, since, as we said in Example 1, our guess of the initial state of LFSR$_s$ was right. By solving the system, we get the initial state of LFSR$_u$. If the guess of the position of the complemented bit in the intercepted sequence is wrong, there is some probability that the system remains consistent if the guess of the initial state of LFSR$_s$ was right. □

Regarding the time complexity of the attack, we should note that it is necessary to check all the possible initial states of LFSR$_s$ in the attack, which gives the time complexity of the attack of $O(c \cdot 2^{L_s})$, where L_s is the length of LFSR$_s$ and c is the number of combinations for complementing the bits of the intercepted sequence. c is small for low levels of noise, which makes the attack feasible in those cases.

IV. Experimental work

The experimental setup involved LFSR$_s$ and LFSR$_u$, both of length 4, with primitive feedback polynomials ($f_s(x) = 1 + x + x^4$ and $f_u(x) = 1 + x^3 + x^4$). 2 positions of the register LFSR$_s$ determined the clocking of LFSR$_u$, i.e. $k = 2$, which means that up to 3 bits of the output sequence of LFSR$_u$ without decimation could be discarded at a time. The length of the intercepted output sequence was 20 and up to 2 bits of the output sequence of the BRM could be complemented by the noise, i.e. $p(1) \leq 10\%$ in the noise sequence. The experiment consisted of the following: for a fixed number of bits of the intercepted sequence complemented by noise t, $t = 1, 2$, all the possible combinations of positions of bits complemented by noise were tried. For each such combination, after complementing the intercepted sequence bits accordingly, the new LCT attack was run and the number of cases in which the guess of the initial state of LFSR$_s$ was right and the consistent system was obtained was recorded. The number of cases in which the guess of the initial state of LFSR$_s$ was wrong and a consistent system was obtained (false alarms) was also recorded. The cases where the number of false alarms was greater than the number of correct guesses followed by consistent systems were of particular interest, since in such a case the solution of the cryptanalytic problem given by our algorithm would be wrong. The goal of the experimental work was to investigate how the number of such cases behaves when t increases. In addition to the number of false alarms, the number of initial states of LFSR$_s$ that the LCT-based attack algorithm would label as solution states would be important to study since in the case of a false alarm, the cryptanalyst would have to check those states further in order to eliminate the wrong solutions. We should bear in mind that the right solution is always offered by the attack algorithm, even when we get false alarms. The experimental results are given in Table I. In that table, for $t = 1$ and $t = 2$, the numbers of false alarms n_f are listed. In addition, the maximum numbers of solutions (i.e. the initial states of LFSR$_s$ offered by the attack

TABLE I
THE NUMBERS OF FALSE ALARMS OBTAINED WITH THE NEW LCT ATTACK (SEE TEXT)

	N	n_f	$n_f\%$	n_s	f_s	$f_s\%$
$t=1$	20	1	5	2	1	5
$t=2$	190	62	33	5	2	1

algorithm) n_s are given together with the numbers of cases f_s, in which the false alarms were generated with the maximum number of solutions (that would be the worst possible case for the cryptanalyst). In the table, N represents the total number of possible combinations for complementing the bits of the intercepted sequence.

From the Table I it can be noted that for $t = 2$ the number of false alarm cases is quite high, approx. 33% of all the cases. The maximum number of solution initial states of LFSR$_s$ offered by the attack algorithm in that case is $n_s = 5$, which is $1/3$ of the possible number of initial states of that register. This number is also quite high, but to recover from such a situation, ordering of these solution states according to the corresponding numbers of consistent systems obtained in the attack is possible to perform, which in most cases places the right initial state to the second position. This eliminates the problem of too many solutions offered by the attack algorithm and also makes the problem of too many false alarms easier. The greatest advantage of the new attack compared with the attack from [1] and [6] is in the fact that the new algorithm always produces the right solution, even when it is accompanied by a false alarm.

V. CONCLUSION

In this paper, a new Linear Consistency Test (LCT)-based attack against a noised pseudorandom generator scheme employing irregular clocking is described and analyzed. The attack was applied against a specific representative of this class of generators known as The Binary Rate Multiplier (BRM). The attack first assigns a system of linear equations to the BRM based on a guessed initial state of its clocking LFSR. This system is then checked for consistency and if consistent, the right initial state of the clocking LFSR was guessed. If the obtained system is inconsistent, the equations involving complemented bits of the intercepted sequence are eliminated from the system and then the consistency of the system is checked again. Which bits of the intercepted sequence are complemented is also guessed. If the guess of those bits is right, the obtained system will surely become consistent. Otherwise, if the guess of the initial state of the clocking LFSR was right, there is a significant chance that the newly obtained system becomes consistent. In a contrary case, the new system will be inconsistent with high probability. The attack always gives the right solution for the initial state of the clocking LFSR, but that solution may be accompanied by other solutions (false alarms). The recovery procedure in those cases is proposed as well. The attack is feasible if the number of possible combinations for the bits of the intercepted sequence complemented by noise is small, which means that the level of noise is up to moderate.

REFERENCES

[1] G. Bu, "Linear consistency test (LCT) in cryptanalysis of irregularly clocked LFSRs in the presence of noise, " Master thesis, Gjøvik University College, Gjøvik, Norway, 2011.
[2] W. Chambers and S. Jennings, "Linear equivalence of certain BRM shift-register sequences, " *Electronics Letters*, vol. 20, no. 24, pp. 1018–1019, 1984.
[3] J. Golić and M. Mihaljević, "A generalized correlation attack on a class of stream ciphers based on the Levenshtein distance, " *Journal of Cryptology*, vol. 3, no. 3, pp. 201–212, 1991.
[4] T. Johansson, "Reduced complexity correlation attacks on two clock-controlled generators, " in: Ohta K. (Ed.), *Advances in Cryptology: Proceedings of ASIACRYPT '98*, Lecture Notes in Computer Science LNCS 1514, pp. 342–356, Springer–Verlag, 1998.
[5] H. Molland, T. Helleseth, "An improved correlation attack against irregular clocked and filtered keystream generators, " in *Proceedings of CRYPTO 2004*, Lecture Notes in Computer Science LNCS 3152, pp. 373–389, Springer–Verlag, 2004.
[6] S. Petrović, "Application of linear consistency test in a ciphertext-only attack on irregularly clocked linear feedback shift registers, " in *Proceedings of XII Spanish Conference on Cryptography and Information Security (RECSI2012)*, U. Zurutuza, R. Uribeetxeberria, I. Arenaza-Nuño, Eds. Arrasate - Mondragon: Servicio Editorial de Mondragon Unibertsitatea, pp. 113-117, 2012.
[7] K. Zeng, C. Yang, and T. Rao, "On the linear consistency test (LCT) in cryptanalysis with applications, " in *Advances in Cryptology, Proceedings of CRYPTO '89*, Lecture Notes in Computer Science LNCS 435, pp. 164–174, Springer–Verlag, 1990.

Modelización lineal de los generadores shrinking a través de las leyes 102 y 60

Sara D. Cardell
Departament d'Estadística
i Investigació Operativa
Universitat d'Alacant
Email: s.diaz@ua.es

Amparo Fúster Sabater
Instituto de Tecnologías Físicas
y de la Información
C.S.I.C.
Email: amparo@iec.csic.es

***Resumen*—En este trabajo se presenta la modelización lineal de los generadores *shrinking* y auto-*shrinking* a través de autómatas celulares lineales uniformes utilizando la ley 102 (ó la 60). La linealidad de estos autómatas se puede utilizar para el criptoanálisis de estos generadores de secuencias.**

***Palabras clave*—Autómatas celulares, criptoanálisis, generadores *shrinking*.**

I. Introducción

En la actualidad, los cifradores en flujo son los procedimientos de cifrado más rápidos que existen, por lo tanto, se utilizan en numerosas aplicaciones tecnológicas, como puede ser el algoritmo A5 para la telefonía en el GSM (véase [1]), el algoritmo E0 para Bluetooth (las especificaciones de Bluetooth se pueden ver en [2]) o el generador J3Gen para etiquetas RFID de bajo coste [3]. A través de una clave corta secreta y un algoritmo público (el generador de la secuencia), un cifrador en flujo genera una secuencia pseudoaleatoria, la secuencia cifrante. Para cifrar nuestro mensaje, éste se suma mediante operaciones XOR con la secuencia cifrante, obteniendo de este modo el texto cifrado. Para recuperar el mensaje inicial, simplemente hay que volver a ejecutar una operación XOR entre la secuencia cifrante y el texto cifrado. De aquí, se deduce la importancia de que la clave sea secreta y conocida únicamente entre las dos partes que comparten la información.

Muchos generadores de secuencias cifrantes utilizan LFSR (Linear Feedback Shift Registers) [4] de máxima longitud combinados con una función booleana no lineal. Existen otros tipos de generadores de secuencias muy populares en criptografía. Todos ellos producen secuencias de cifrado con una complejidad lineal alta, largo periodo y buenas propiedades estadísticas.

Por otro lado, se ha probado que algunos autómatas lineales de una dimensión generan exactamente las mismas secuencias que un LFSR de máxima longitud. Por lo tanto, un autómata puede ser considerado un generador alternativo a un LFSR de longitud máxima [5]. Además, algunos generadores criptográficos diseñados a través de varios LFSR pueden ser modelados como autómatas celulares lineales. En [5], [6], los autores modelizaron los generadores *shrinking* y auto-*shrinking* usando las leyes 150 y 90. La idea principal de este trabajo es modelizar esos mismos generadores utilizando las leyes 102 y 60. Estas leyes se han utilizado previamente, junto con la ley 90, en otros ámbitos, como la construcción del triángulo de Sierpinski [7], [8]. También es posible encontrar ambas leyes en el análisis de autómatas celulares complementados derivados del grupo de autómatas lineales híbridos [9] o como parte de métodos propuestos para generar familias de grafos expandidos a través de autómatas con frontera nula [10].

II. Preliminares

En esta sección, se presentan algunas definiciones necesarias para la comprensión del resto del trabajo. En la primera subsección se introducen las definiciones de generador *shrinking* y generador auto-*shrinking*. En la segunda subsección, el concepto de autómata celular es recordado.

II-A. Generadores

El **generador *shrinking*** consta de dos LFSR, R_1 y R_2, tales que la secuencia $\{a_i\}$ producida por el primer registro R_1 decima la secuencia $\{b_i\}$ producida por el segundo registro R_2. La secuencia de salida del generador $\{s_j\}$ se obtiene del siguiente modo:

$$\begin{cases} \text{Si } a_i = 1 \text{ entonces } s_j = b_i. \\ \text{Si } a_i = 0 \text{ entonces } b_i \text{ es rechazado.} \end{cases}$$

Si L_1 y L_2 son las longitudes de R_1 y R_2, respectivamente, el período de la secuencia $\{s_j\}$ es $T = (2^{L_2} - 1)2^{L_1 - 1}$, siempre que L_1 y L_2 sean primos entre sí. A su vez, la complejidad lineal, denotada por LC cumple $L_2 2^{L_1 - 2} < LC \leq L_2 2^{L_1 - 1}$. Además, es una secuencia casi equilibrada, ya que el número de unos en dicha secuencia viene dado por $2^{L_1 + L_2 - 2}$. El generador *shrinking* tiene buenas propiedades criptográficas y es fácil de implementar [11], por lo tanto, es adecuado para su implementación en sistemas de cifrado en flujo.

Por otro lado, el **generador auto-*shrinking*** fue diseñado por Meier y Staffelbach para uso en aplicaciones de cifrado en flujo [12]. Es bastante atractivo, debido a su simplicidad ya que implica el uso de un solo LFSR. Este generador consiste en un LFSR de máxima longitud que produce una secuencia que es decimada por ella misma, por lo tanto, es un caso simplificado y concreto del generador *shrinking*. La regla de decimación es bastante simple; dado un par de bits consecutivos $\{a_{2i}, a_{2i+1}\}$ de la secuencia $\{a_i\}$ generada por el LFSR, la secuencia de salida $\{s_j\}$ se obtiene del siguente modo:

$$\begin{cases} \text{Si } a_{2i} = 1 \text{ entonces } s_j = a_{2i+1}. \\ \text{Si } a_{2i} = 0 \text{ entonces } a_{2i+1} \text{ es rechazado.} \end{cases}$$

El periodo de la secuencia viene dado por $T = 2^{L-1}$ [12], siendo L la longitud del LFSR que genera la secuencia de entrada del generador. Además, la complejidad lineal, denotada por LC, cumple $2^{L-2} < LC \leq 2^{L-1} - (L-2)$ [13].

Para ambos generadores, la clave es el estado inicial del LFSR y el polinomio de realimentación, también recomendado como parte de la clave.

II-B. Autómatas Celulares

Un **autómata celular** (CA) de una dimensión es un registro compuesto de n celdas cuyo contenido (binario en este trabajo) se actualiza de acuerdo a una ley o función de k variables [14]. Así, el estado de la celda que hay en la posición i en el instante $t+1$, x_i^{t+1}, depende del estado de las k celdas vecinas en el instante t. Si estas leyes se componen exclusivamente de operaciones XOR, entonces el autómata se dice que es **lineal**.

En un autómata, todas las celdas pueden obedecer la misma ley; en ese caso el autómata se dice que es **uniforme** o **regular**. Si obedecen distintas leyes, se dice que es **híbrido**. Por otro lado, se dice que el autómata tiene **frontera nula** o que es **nulo**, si se consideran nulos los valores adyacentes a las celdas de los extremos. Si, por el contrario, las celdas de los extremos se consideran adyacentes, se dice que el autómata tiene **frontera cíclica** o que es **periódico**.

En este trabajo, se consideran solamente autómatas uniformes de una dimensión tanto nulos como periódicos con las leyes 102 y 60. Para $k = 3$, estas leyes vienen dadas por:

Ley 102: $x_i^{t+1} = x_i^t + x_{i+1}^t$

111	110	101	100	011	010	001	000
0	1	1	0	0	1	1	0

Ley 60: $x_i^{t+1} = x_{i-1}^t + x_i^t$

111	110	101	100	011	010	001	000
0	0	1	1	1	1	0	0

Los números 01100110 y 00111100 son las representaciones binarias de 102 y 60, respectivamente. De ahí que sean llamadas ley 102 y ley 60.

La idea principal de este trabajo es proporcionar autómatas celulares lineales uniformes, donde una de las secuencias de salida corresponde a la secuencia generada por los generadores *shrinking*. La finalidad de esta modelización es expresar una secuencia pseudoaleatoria no lineal en términos de un modelo lineal, para facilitar el posible criptoanálisis de estas secuencias.

III. Modelización

Las secuencias producidas por LFSR de longitud máxima gozan de buenas propiedades criptográficas, pero sufren del *mal de la linealidad*. Gracias al algoritmo de Belekamp-Massey [15], si se intercepta como mínimo un número de bits igual al doble de la complejidad lineal de la secuencia de salida de un LSFR, ésta puede ser recuperada totalmente. Por esta razón, se introdujo el uso de funciones booleanas no lineales que rompieran esa linealidad.

Los generadores *shrinking* fueron introducidos para romper esta linealidad, sin embargo, se puede comprobar en las secciones III-A y III-B, que las secuencias producidas por estos generadores, pueden ser obtenidas a su vez como salida de una estructura lineal, autómatas celulares uniformes lineales en este caso. Esto implica que las secuencias sean sensibles a sufrir un criptoanálisis que explote esta linealidad.

III-A. Modelizando el generador shrinking

Sea $\mathbb{F}_2$ el cuerpo de Galois de dos elementos. Dados dos polinomios primitivos $p_1(x), p_2(x) \in \mathbb{F}_2[x]$, cuyos grados son L_1 y L_2, respectivamente, primos entre sí, el periodo de la secuencia de salida del generador es $T = (2^{L_2} - 1)2^{L_1 - 1}$.

Existe un autómata celular uniforme periódico de longitud T que genera la secuencia de salida utilizando la ley 102. En algunos casos, sin embargo, la longitud del autómata se ve reducida hasta llegar a ser un divisor de T. Veamos el siguiente ejemplo.

Ejemplo 1: Dados los polinomios primitivos $p_1(x) = 1+x+x^2$ y $p_2(x) = 1+x+x^3$, la secuencia generada por el generador *shrinking* tiene periodo $T = (2^3 - 1)2^{2-1} = 14$. Si se toman como estados iniciales 10 y 100, respectivamente, la secuencia de salida del generador *shrinking* es 10111000110101. En la Tabla I podemos ver un ejemplo de autómata uniforme periódico que genera la secuencia dada. Si se considera el autómata celular uniforme periódico de longitud 14 que utiliza la ley 60, en vez de la ley 102, la secuencia buscada aparecería en último lugar en vez de en primer lugar como pasa en el autómata celular de la Tabla I. De hecho, el resto de secuencias serían las mismas pero aparecerían en orden inverso. □

Diversas simulaciones para distintos valores de las longitudes L_1 y L_2 se han llevado a cabo utilizando Matlab. Para todos los casos en los que la longitud del autómata celular que genera la secuencia del generador *shrinking* es el T y no un divisor de éste, es posible observar que aparecen 2^{L_1-1} secuencias diferentes repetidas $2^{L_2} - 1$ veces y todas ellas con periodo T.

III-B. Modelizando el generador auto-shrinking

Dado un polinomio primitivo $p(x) \in \mathbb{F}_2[x]$ de grado L, sabemos que el periodo de la secuencia de salida del generador *auto-shrinking* es 2^{L-1}.

Para esta secuencia con periodo $T = 2^{L-1}$, existe un autómata celular uniforme nulo que la genera utilizando la ley 102. Observando las simulaciones obtenidas en Matlab para diversos valores de L, se deduce en todos los casos que la longitud del autómata es exactamente la complejidad lineal de la secuencia dada. Véase el siguiente ejemplo.

Tabla I
AUTÓMATA CELULAR QUE GENERA LA SECUENCIA CONSIDERADA EN EL EJEMPLO 1

102	**102**	**102**	**102**	**102**	**102**	**102**	**102**	**102**	**102**	**102**	**102**	**102**	**102**
1	1	0	1	0	0	1	0	0	1	1	0	1	1
0	1	1	1	0	1	1	0	1	0	1	1	0	0
1	0	0	1	1	0	1	1	1	1	0	1	0	0
1	0	1	0	1	1	0	0	0	1	1	1	0	1
1	1	1	1	0	1	0	0	1	0	0	1	1	0
0	0	0	1	1	1	0	1	1	0	1	0	1	1
0	0	1	0	0	1	1	0	1	1	1	1	0	1
0	1	1	0	1	0	1	1	0	0	0	1	1	1
1	0	1	1	1	1	0	1	0	0	1	0	0	1
1	1	0	0	0	1	1	1	0	1	1	0	1	0
0	1	0	0	1	0	0	1	1	0	1	1	1	1
1	1	0	1	1	0	1	0	1	1	0	0	0	1
0	1	1	0	1	1	1	1	0	1	0	0	1	0
1	0	1	1	0	0	0	1	1	1	0	1	1	0

Tabla II
AUTÓMATA CELULAR QUE GENERA LA SECUENCIA CONSIDERADA EN EL EJEMPLO 2

102	**102**	**102**	**102**	**102**
0	0	0	1	1
0	0	1	0	1
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	1	1	1	1
0	0	0	0	1

Ejemplo 2: Dado el polinomio primitivo $p(x) = 1 + x + x^4$, la secuencia generada por el generador auto-*shrinking* tiene periodo $T = 2^3 = 8$. Si se toma el estado inicial 1001, la secuencia de salida del generador es 00011110. El polinomio característico de esta secuencia es $P_M(x) = (1 + x)^5$, por lo que la complejidad lineal será cinco. En la Tabla II se proporciona un ejemplo de autómata uniforme nulo de longitud cinco que genera la secuencia dada. Como sucedía en el ejemplo 1, si se considera el autómata celular uniforme nulo de longitud 5 que utiliza la ley 60, en vez de la ley 102, las secuencias de salida serían las mismas que en el autómata celular de la Tabla II pero aparecerían en orden inverso. □

III-C. Comparación con otros modelos lineales basados en autómatas celulares

En [5], [6], otros modelos basados en autómatas celulares híbridos y nulos fueron propuestos. En este caso, los autores propusieron autómatas celulares basados en las leyes 150 y 90, que también generaban las secuencias de salida de los generadores *shrinking*.

III-C1. Generador shrinking: En el caso del generador *shrinking*, las secuencias de salida tienen como polinomio característico $P_M(x) = P(x)^p$, con $P(x) \in \mathbb{F}_2[x]$ un polinomio de grado L_2 y p un entero tal que $2^{L_1-2} < p \leq 2^{L_1-1}$, donde L_1 y L_2 son los grados de los polinomios de realimentación de los LFSR que generan las secuencias de entrada para el generador. En [5], un algoritmo basado en la concatenación de autómatas y en el algoritmo de Cattell y Muzio [16] es propuesto. Este algorimto proporciona autómatas celulares nulos de longitud $L_2 2^{L_1-1}$ basados en las leyes 150 y 90, que generan las mismas secuencias que el generador *shrinking*.

Estos autómatas tienen una longitud notablemente inferior a la longitud de los autómatas considerados en la sección III-A. Los autómatas propuestos en este trabajo tienen una longitud igual al periodo de la secuencia, $T = (2^{L_2} - 1)2^{L_1-1}$, frente a la longitud $L_2 2^{L_1-1}$ de los autómatas considerados en [5]. Sin embargo, en los autómatas propuestos en la sección III-A aparecen 2^{L_1-1} secuencias repetidas $2^{L_2} - 1$ veces, por lo que, conociendo las primeras 2^{L_1-1} secuencias, se deduce el resto del autómata celular. Por lo tanto, tienen un comportamiento más predecible que puede ser de ayuda para recuperar la secuencia, dada una cantidad de bits interceptada. En la Tabla I, se puede observar que la primera secuencia es la misma que la tercera secuencia después de haber sufrido un traslación cíclica de cuatro posiciones. A su vez, la quinta secuencia es la misma que la segunda secuencia después de haber sufrido una traslación cíclica de cuatro posiciones también. En total, la misma secuencia aparece siete veces pero trasladada. Lo mismo ocurre con la segunda secuencia; aparece siete veces después de sufrir traslaciones de cuatro posiciones. Cuando la longitud del autómata es un divisor de T, tenemos la misma cantidad de secuencias 2^{L_1-1}, pero se repiten una cantidad de veces inferior a $2^{L_2} - 1$; hecho favorable a la hora de realizar un posible criptoanálisis.

Por otro lado, para obtener los autómatas celulares pro-

puestos en [5], había que aplicar el algoritmo de Cattell y Muzio [16] y había que aplicar, posteriormente, una concatenación. En este caso, con saber el periodo de la secuencia conocemos el autómata, ya que éste será un autómata uniforme que utilizará la ley 102 (ó la 60) en todas sus celdas, y tendrá longitud T.

III-C2. Generador auto-shrinking: Para el generador auto-*shrinking*, al ser un caso específico del generador *shrinking*, el polinomio característico es de la forma $P_M = (1+x)^p$, con $2^{L-2} < p \leq 2^{L-1}$, donde L es la longitud del LFSR que genera la secuencia de entrada para el generador. En este caso existe un autómata celular nulo de longitud 2^{L-1} basado en las leyes 150 y 90 que genera las mismas secuencias que el generador auto-*shrinking*. Los autómatas celulares obtenidos tienen una estructura definida; se considera siempre la ley 90 en las celdas de los extremos y la ley 150 en el resto de celdas (véase [6]).

Los autómatas celulares propuestos en este trabajo, poseen también una estructura definida. Para los que utilizan la ley 102, aparece siempre la secuencia de unos en último lugar y en penúltimo lugar aparece siempre una secuencia de periodo dos (la secuencia $0101\ldots$ o la secuencia $1010\ldots$). Después hay dos secuencias de periodo cuatro, cuatro secuencias de periodo ocho y, así sucesivamente, hasta encontrar 2^{L-3} secuencias de periodo 2^{L-2}. El resto de secuencias (la longitud del autómata menos 2^{L-2}) son de periodo 2^{L-1}, incluida la secuencia de salida del generador auto-*shrinking*. Por otro lado, sabemos que la complejidad lineal cumple $2^{L-2} < LC \leq 2^{L-1} - (L-2)$, por lo tanto, la longitud de estos autómatas (exactamente LC) es menor que 2^{L-1}, la longitud de los autómatas propuestos en [6], con lo que se reduce la complejidad de computar estos modelos.

En la Tabla II, podemos observar que de las cinco secuencias, existe una secuencia de periodo uno, la secuencia de unos, en el último lugar. También aparece la secuencia de periodo dos en el penúltimo lugar y otras dos secuencias de periodo cuatro. Por último aparece una sola secuencia de periodo ocho, la secuencia de salida del generador auto-*shrinking*. Para modelizar esta misma secuencia utilizando las leyes 90 y 150, se necesita un autómata celular de longitud seis (véase [6]).

IV. Aplicaciones

La principal aplicación de modelizar las secuencias de salida de este tipo de generadores es el criptoanálisis, ya que estos generadores tienen buenas propiedades criptográficas y son adecuados para ser usados en cifrado en flujo. Dado un modelo lineal que describe el comportamiento del generador, el criptoanálisis puede llevarse a cabo utilizando diferentes herramientas.

- En primer lugar, se puede hacer una búsqueda exhaustiva entre todos los estados iniciales posibles del autómata celular. En el caso del generador *shrinking*, si los grados de los dos polinomios que se utilizan para generar las secuencias de entrada son L_1 y L_2, respectivamente, el autómata tendría longitud $(2^{L_2}-1)2^{L_1-1}$. La complejidad computacional de probar todos los estados sería proporcional a $2^{(2^{L_2}-1)2^{L_1-1}}$. Sin embargo, en este autómata aparecen 2^{L_1-1} secuencias repetidas $2^{L_2}-1$ veces, por lo que sólo sería necesario generar las 2^{L_1-1} primeras secuencias. Una vez generadas estas secuencias, se puede deducir el resto de las secuencias. La complejidad ascendería a $2^{2^{L_1-1}}$. En este caso, sería más eficiente hacer una búsqueda exhaustiva entre los posibles estados iniciales de los LFSR, ya que habría que considerar $2^{L_1+L_2}$ casos. Para el generador auto-*shrinking*, sucede algo parecido. Sea L la longitud del LFSR que genera la secuencia de entrada para este generador. Si se quisieran probar todos los estados iniciales del LFSR, el número de estados asciende a 2^L. Por otro lado, la longitud del autómata LC, cumple $2^{L-2} < LC \leq 2^{L-1} - (L-2)$. Por lo tanto, el número total de estados iniciales del autómata que se tendrían que probar 2^{LC}, es mayor que $2^{2^{L-2}}$ y, a su vez, $2^{2^{L-2}} > 2^L$ para $L > 4$.
- Por otro lado, si se recupera una cantidad de bits consecutivos igual a $2^{L_1-1}+t$, siento t la longitud de la traslación cíclica que sufren las secuencias obtenidas por los autómatas que generan la secuencia de salida del generador *shrinking*, podemos recuperar la secuencia entera. Es posible ver que esta traslación se puede obtener dividiendo la longitud del autómata entre el número de secuencias diferentes, esto es, 2^{L_1-1}. Por lo tanto, con 2^{L_1} bits consecutivos es posible recuperar la totalidad de la secuencia. Nótese que esta cantidad es menor que la complejidad lineal de la secuencia, véase la sección II-A. Para el generador auto-*shrinking*, basta interceptar una cantidad de bits igual a la complejidad lineal de la secuencia para recuperar la totalidad de ésta. En ambos casos cabe resaltar que la complejidad lineal es la mitad de la cantidad de bits que se necesitan para llevar a cabo el algoritmo de Berlekamp-Massey [15].
- Es posible combinar los autómatas celulares propuestos en este trabajo y los autómatas celulares propuestos en [5], [6] con les leyes 150 90, para recuperar pequeñas cantidades de bits de las secuencias del autómata, que pueden ayudar a recuperar la totalidad de la secuencia. Por ejemplo, cuando el autómata que modeliza el generador *shrinking* comienza con la ley 150, las dos primeras secuencias de estos autómatas y los propuestos en la sección II-A, son las mismas. Por lo que recuperar una parte de una de las secuencias en uno de los autómatas nos lleva a recuperar otra parte en el otro autómata.

V. Conclusión

Los esfuerzos por parte de los criptógrafos de incluir generadores por decimación con la finalidad de romper la linealidad de las secuencias generadas por LFSR han sido inútiles, ya que, las secuencias de salida de estos generadores pueden modelizarse como secuencias de salida de estructuras lineales. Este trabajo analiza una familia de autómatas celulares uniformes lineales, basados en la ley 102 (60) que describen el comportamiento de los generadores *shrinking*, diseñados como no lineales.

AGRADECIMIENTOS

El trabajo del primer autor ha sido financiado por una beca postdoctoral de la Generalitat Valenciana con referencia APOSTD/2013/081 y por el proyecto MTM2011-24858 del Ministerio de Ciencia e Innovación del Gobierno de España.

El trabajo del segundo autor ha sido financiado por el Ministerio de Ciencia e Innovación del Gobierno de España bajo el proyecto "TUERI: Technologies for secure and efficient wireless networks within the Internet of Things with applications to transport and logistics", TIN2011-25452.

REFERENCIAS

[1] GSM, Global Systems for Mobile Communications, http://cryptome.org/gsm-a512.htm

[2] Bluetooth, Specifications of the Bluetooth system, http://www.bluetooth.com

[3] A. Peinado, J. Munilla, y A. Fúster-Sabater, "EPCGen2 Pseudorandom Number Generators: Analysis of J3Gen," *Sensors*, vol. 14, no. 4, pp. 6500–6515, 2014.

[4] S. W. Golomb, *Shift Register-Sequences*. Laguna Hill, California: Aegean Park Press, 1982.

[5] A. Fúster-Sabater y P. Caballero-Gil, "Linear solutions for cryptographic nonlinear sequence generators," *Physics Letters A*, vol. 369, pp. 432–437, 2007.

[6] A. Fúster-Sabater, M. E. Pazo-Robles, y P. Caballero-Gil, "A simple linearization of the self-shrinking generator by means of cellular automata," *Neural Networks*, vol. 23, no. 3, pp. 461–464, 2010.

[7] S. Wolfram, "Computation theory of cellular automata," *Communications in Mathematical Physics*, vol. 96, no. 1, pp. 15–57, 1984.

[8] S. Wolfram, *A new kind of science*. Wolfram-Media, 2002.

[9] S. Cho, U. Choi, H. Kim y Y. Hwang, "Analysis of complemented CA derived from linear hybrid group CA," *Computers and Mathematics with Applications*, vol. 53, no. 1, pp. 54–63, 2007.

[10] S. Cho, U. Choi, H. Kim, Y. Hwang y J. Kim, "60/102 Null Boundary Cellular Automata based expander graphs," in *16th Intl. Workshop on CA and DCS*. Discrete Mathematics and Theoretical Computer Science (DMTCS) Proceedings, 2010, pp. 19–28.

[11] D. Coppersmith, H. Krawczyk, y Y. Mansour, "The shrinking generator," in *Advances in Cryptology – CRYPTO '93*, ser. Lecture Notes in Computer Science. Springer-Verlag, 1993, vol. 773, pp. 23–39.

[12] W. Meier y O. Staffelbach, "The self-shrinking generator," in *Advances in Cryptology – EUROCRYPT 1994*, ser. Lecture Notes in Computer Science. Springer-Verlag, 1994, vol. 950, pp. 205–214.

[13] S. R. Blackburn, "The linear complexity of the self-shrinking generator," *IEEE Transactions on Information Theory*, vol. 45, no. 6, pp. 2073–2077, 1999.

[14] A. K. Das, A. Ganguly, A. Dasgupta, S. Bhawmik, y P. P. Chaudhuri, "Efficient characterisation of cellular automata," *IEE Proceedings E: Computers and Digital Techniques*, vol. 137, no. 1, pp. 81–87, 1990.

[15] J. L. Massey, "Shift-register synthesis and BCH decoding," *IEEE Transactions on Information Theory*, vol. 15, no. 1, pp. 122–127, 1969.

[16] K. Cattell y J. C. Muzio, "One-dimensional linear hybrid cellular automata," *IEEE Transactions on Computer-Aided Design*, vol. 15, no. 3, pp. 325–335, 1996.

Calculando Equivalentes Débiles de Filtrados No Lineales

A. Fúster-Sabater
Inst. Tecnologías Físicas y de la Información
Consejo Superior Investigaciones Científicas
Email: amparo@iec.csic.es

P. Caballero-Gil
Departamento de Informática
Universidad de La Laguna
Email: pcaballe@ull.edu.es

***Resumen*—Dada una secuencia binaria generada con una función filtro no lineal aplicada sobre un registro de desplazamiento lineal o LFSR (Linear Feedback Shift Register), siempre es posible generar la misma secuencia a partir de cualquier otro LFSR de la misma longitud mediante el uso de otra función filtro. El problema aún sin resolver es el cálculo de la función filtro equivalente para cada LFSR. En este trabajo se analiza el caso en el que se utiliza un LFSR recíproco para generar un equivalente de un filtrado no lineal de partida mediante el cálculo de la relación específica entre ambas funciones filtro. Además, desde un punto de vista criptográfico, el método aquí desarrollado permite determinar filtrados equivalentes inseguros de otros que son aparentemente seguros. Este resultado puede considerarse como una demostración de que, para que un generador de secuencia cifrante pueda llegar a ser considerado totalmente seguro, debe cumplir diferentes propiedades, algunas de las cuales son aún desconocidas.**

***Palabras clave*—Criptografía de clave secreta (*Secret key cryptography*), Generador pseudoaleatorio (*Pseudorandom generator*), Cifrado en flujo (*Stream cipher*), Filtrado no lineal, LFSR.**

I. Introducción

Un cifrado en flujo está compuesto por un generador de secuencia cifrante cuya secuencia pseudoaleatoria de salida se suma módulo 2 con los bits del texto en claro. Dado que la operación de cifrado es muy rápida, se considera que los cifrados en flujo son en general más eficientes que cualquier otro tipo de cifrado. Esta es la razón principal por la que resultan especialmente adecuados para las comunicaciones inalámbricas como las de telefonía móvil, Wi-Fi o Bluetooth.

Una de las formas más habituales de construir generadores de secuencia cifrante es mediante el uso de un generador pseudoaleatorio conocido como registro de desplazamiento lineal o LFSR (Linear Feedback Shift Register) [9], cuya secuencia de salida es la imagen de una función lineal aplicada sobre sus estados sucesivos. Si se cumplen determinadas condiciones, esta estructura produce secuencias con características muy deseables para uso criptográfico. En particular, si su polinomio característico es primitivo, la secuencia generada, llamada *m*-secuencia, tiene algunas propiedades muy útiles, tales como un período grande y una buena distribución estadística de ceros y unos. Sin embargo, la secuencia producida por un LFSR nunca debe utilizarse como secuencia cifrante en un cifrado en flujo porque la linealidad inherente de la estructura podría ser fácilmente utilizada para romper el cifrado.

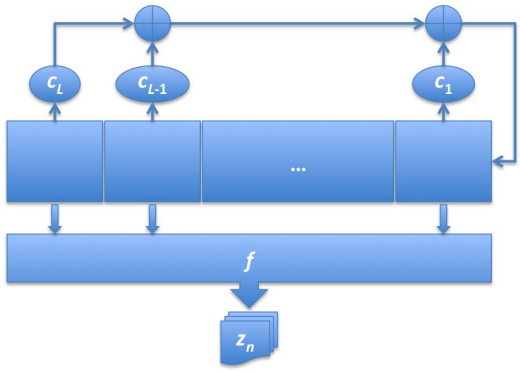


Figura 1. Filtrado No Lineal

Un interesante generador de secuencia cifrante basado en un LFSR es el filtrado no lineal, que produce una secuencia cifrante a partir de la salida de una función booleana no lineal aplicada sobre los estados de un LFSR. En particular, un filtrado no lineal consta de dos partes (ver Fig. 1):

1. Un LFSR de longitud L, con polinomio característico $P(x) = x^L + c_1 \cdot x^{L-1} + \cdots + c_{L-1} \cdot x + c_L$ de coeficientes binarios que, a partir de un estado inicial IS (Initial State), genera una secuencia de salida $\{a_n\}$.
2. Una función booleana no lineal $F : GF(2)^L \rightarrow GF(2)$, llamada función filtro, cuyas variables de entrada son los L bits de los sucesivos estados del LFSR, y cuya imagen es la secuencia binaria $\{Z_n\}$.

Aunque las secuencias producidas por LFSR están bien estudiadas, no puede decirse lo mismo de las secuencias obtenidas con filtrados no lineales.

Este trabajo trata sobre la relación entre los diferentes filtrados no lineales que producen exactamente la misma secuencia. El objetivo principal es mostrar que, aunque el estudio de las propiedades de un generador puede llevar a la conclusión de que las propiedades de la secuencia generada son buenas, a veces esa deducción puede ser errónea. En particular, este trabajo muestra que dos estructuras con niveles de seguridad aparentemente diferentes pueden producir la misma secuencia cifrante. De hecho, este resultado puede verse como una demostración de que el nivel de seguridad real de un generador es siempre el nivel de seguridad del elemento más débil de la clase de generadores equivalentes.

La organización de este trabajo es de la siguiente forma. La sección II incluye una breve revisión de algunos trabajos relacionados. En la sección III, tras los preliminares necesarios, se aborda el problema del recuento general de filtrados no lineales equivalentes, así como un estudio de la relación entre ellos. Después, la sección IV presenta una breve explicación de la propuesta, que está basada en el nuevo concepto de filtrados recíprocos, e introduce un nuevo método para el cálculo de equivalentes débiles de filtrados no lineales mediante un ejemplo didáctico. Finalmente, la sección V describe algunas conclusiones y posibles líneas futuras de investigación.

II. Trabajos Relacionados

Una herramienta bastante útil para estudiar secuencias binarias es el algoritmo de Berlekamp-Massey [12], que determina el LFSR más corto que permite generar cualquier secuencia binaria finita de entrada. La longitud de dicho LFSR se conoce como complejidad lineal de la secuencia. Algunas cotas generales, tanto inferiores como superiores, de la complejidad lineal de las secuencias filtradas han sido publicadas en [10] y [5], mientras que cotas más ajustadas pero para casos específicos pueden encontrarse en [16] y [17].

El proyecto eSTREAM [3] representa el esfuerzo más importante en cuanto al diseño de cifrados en flujo. Fue un proyecto de varios años que tenía como objetivo promover el diseño de sistemas de cifrado en flujo eficientes adecuados para su adopción generalizada. Como resultado fueron escogidos siete generadores de secuencia cifrante con dos perfiles diferentes, software y hardware. Uno de ellos, el llamado SOSEMANUK, es un generador basado en un LFSR donde la longitud del LFSR utilizado es 10 y el contenido de cada etapa es un elemento de $GF(2^{32})$. Dicho generador responde a principios de diseño similares a los del generador SNOW 2.0, predecesor del generador SNOW 3G, que constituye el núcleo de los algoritmos de protección de la confidencialidad y de la integridad en la cuarta generación de comunicaciones de telefonía móvil, LTE y LTE-Advanced, [19].

Existen varias referencias bibliográficas interesantes de ataques criptográficos contra filtrados no lineales.

El primer ataque de correlación básico contra un filtrado no lineal fue publicado en [18]. En él, las correlaciones entre la *m*-secuencia $\{a_n\}$ producida por el LFSR y la secuencia filtrada $\{Z_n\}$ son utilizadas para construir un generador equivalente consistente en una combinación no lineal de varios LFSR. Los principales inconvenientes de ese ataque son la enorme cantidad de tiempo requerido para calcular las correlaciones necesarias, y el requisito de que la función filtro F debe tener una alta correlación con una función afín. Tras definir la no linealidad de una función booleana como la distancia de Hamming mínima entre esa función y una función afín, una consecuencia práctica del ataque de correlación básico es que todo diseñador debe elegir siempre funciones filtro altamente no lineales para los filtrados no lineales. Luego, el concepto subyacente a los ataques de correlación básicos fue mejorado, proponiéndose el ataque de correlación rápido descrito en [13]. Dos desventajas comunes de las diferentes versiones de esos ataques son el gran número de bits de secuencia cifrante interceptados que son necesarios para llevarlos a cabo, y la suposición de que la función filtro no es altamente no lineal.

Un ataque general de inversión fue propuesto en [8] contra cualquier función filtro. Como consecuencia, se obtuvo una caracterización sencilla de los filtrados que son resistentes contra dicho ataque de inversión. Por otra parte, los trabajos [6] y [4] propusieron el llamado ataque por decimación contra cualquier generador de secuencia cifrante basado en un LFSR. La idea es considerar una secuencia decimada de la secuencia cifrante interceptada de manera que la secuencia decimada pueda ser generada a partir de una secuencia generada por el LFSR decimada. Sin embargo, según [16] si la longitud L del LFSR es un número primo, entonces el ataque por decimación no proporciona ninguna ventaja.

En los últimos años se han publicado varios ataques algebraicos contra cifrados en flujo. En ellos, el atacante utiliza los bits de la secuencia interceptada para establecer un sistema no lineal de ecuaciones polinómicas en función de los bits generados por el LFSR. El principal problema con respecto a este tipo de ataques es que, como se muestra en [7], el problema para obtener la solución de un sistema no lineal de ecuaciones multivariantes es NP-duro, incluso si todas las ecuaciones son de segundo grado y el cuerpo subyacente es $GF(2)$. Para hacer frente a este problema, se propuso el conocido como método algebraico XL [2] que permite resolver un sistema no lineal de ecuaciones cuadráticas para algunos filtrados no lineales. Con el fin de incrementar la resistencia frente a este ataque, la función filtro debe ser no sólo altamente no lineal, sino también tener una gran distancia con respecto a aproximaciones de pequeño grado algebraico.

Los ataques basados en el equilibrio tiempo-memoria-datos (time-memory-data tradeoff) [1] pueden evitarse fácilmente en los filtrados no lineales mediante el uso de LFSR de gran longitud. Hay otro ataque interesante, llamado de suposición y determinación (guess and determine) [14], que explota la relación entre los valores internos (tales como la recurrencia lineal en el LFSR), y la relación utilizada para construir la secuencia cifrante a partir de esos valores internos. Como su nombre indica, este ataque trata de adivinar algunos valores internos para luego usar las relaciones mencionadas y determinar otros valores internos. Después de un ataque de ese tipo, el cifrado se considera roto cuando un estado interno completo ha podido ser determinado a partir de valores adivinados. Este tipo de ataque puede evitarse mediante la elección adecuada del polinomio del LFSR.

Uno de los trabajos más estrechamente relacionados con el presente es [11], donde se propuso un ataque por transformación lineal contra un filtrado no lineal. La idea detrás de ese ataque es transformar el generador dado en un filtrado no lineal equivalente con el mismo LFSR pero con una función filtro más adecuada para algunos de los ataques mencionados.

Otro trabajo reciente es [15], donde se define una clase de equivalencia de los filtrados no lineales, demostrando que un número importante de propiedades criptográficas no son invariantes entre los elementos de la misma clase de equiva-

lencia. Los propios autores reconocen que la determinación del cifrado equivalente más débil de dicha clase es una tarea muy difícil debido a que el tamaño de la clase de equivalencia es muy grande. El presente trabajo no estudia dicha clase de equivalencia completa, sino sólo uno de sus elementos, que hemos identificado que en muchos casos conduce a un generador equivalente más débil que el filtrado de partida.

En conclusión, cada ataque contra el filtrado no lineal suele conducir a nuevas conclusiones sobre propiedades deseables del LFSR y/o de la función filtro. En consecuencia, uno de los principales temas de investigación con respecto a los filtrados no lineales es sobre cómo construir una buena función booleana para aumentar la resistencia contra los ataques mencionados. Este trabajo se ocupa de esta cuestión ya que demuestra que las propiedades del generador no siempre garantizan la seguridad de las secuencias producidas.

III. Estudio General de Filtrados Equivalentes

En esta sección se obtiene el número de filtrados equivalentes y luego se analiza la relación entre ellos.

Dado un filtrado no lineal consistente en un LFSR con polinomio característico $P_1(x)$ y función filtro $F_1(x)$, siempre es posible generar la misma secuencia con cualquier otro LFSR de la misma longitud y otra función filtro.

Si α es una raíz del polinomio característico $P_1(x)$ a la vez que un elemento primitivo de $GF(2^L)$ y $(k, 2^L-1) = 1$, entonces se tiene que α^k es también un elemento primitivo de $GF(2^L)$. Por tanto, se concluye que hay $\phi(2^L-1)$ elementos primitivos de $GF(2^L)$. En particular, los L conjugados de cualquier elemento (que son las sucesivas potencias cuadradas), por ejemplo, $\alpha, \alpha^2, \alpha^4, \alpha^8, ..., \alpha^{2^{L-1}}$, son elementos primitivos de $GF(2^L)$ además de raíces del mismo polinomio, que puede calcularse mediante la expresión $\prod_{i=0}^{L-1}(x-\alpha^{2^i})$ en $GF(2^L)$.

Por tanto, hay $\phi(2^L-1)/L$ polinomios primitivos de $GF(2^L)$, cada uno con L raíces que son todos los conjugados de un elemento primitivo. Puesto que cada uno de estos polinomios define un LFSR de longitud L, hay $\Phi(2^L-1)/L$ LFSR diferentes de longitud L, cada uno correspondiente a un conjunto de conjugados de un elemento primitivo de $GF(2^L)$.

En conclusión, dado que cualquier secuencia obtenida con un filtrado no lineal puede ser generada mediante una función filtro sobre cada LFSR, entonces hay $\Phi(2^L-1)/L$ filtrados no lineales diferentes que pueden ser utilizados para generarla.

La relación entre dos elementos primitivos, α y β, raíces de dos polinomios característicos de dos LFSR diferentes de longitud L viene dada por la expresión $\beta = \alpha^k$ siendo $mcd(k, 2^L-1) = 1$ y $k \neq 2^i \cdot$ j (mod 2^L-1) con $i, j > 0$.

Esta información sobre la relación entre los polinomios característicos $P_1(x)$ y $P_2(x)$ de dos LFSR podría ayudar a definir la relación entre las dos funciones filtro $F_1(x)$ y $F_2(x)$ que forman parte de los dos generadores equivalentes que producen la misma secuencia filtrada.

Como se puede ver en la tabla I, los casos $k = 1$ y $k = 2^{L-1}-1$ siempre determinan diferentes conjuntos de raíces conjugadas que definen diferentes LFSR. De hecho, los polinomios correspondientes a las raíces α y $\beta = \alpha^{2^{L-1}-1}$ son siempre recíprocos.

Tabla I
Ejemplos de recuento de filtrados equivalentes

L	3	4	5	6
2^L-1	7	15	31	63
N. filtrados	2	2	6	6
k por filtro	1,2,4	1,2,4,8	1,2,4,8,16	1,2,4,8,16,32
	3,5,6	7,11,13,14	3,6,12,24,17	5,10,20,40,17,34
			5,10,20,9,18	11,22,44,25,50,37
			7,14,28,25,19	13,26,52,41,19,38
			11,22,13,26,21	23,46,29,58,53,43
			15,30,29,27,23	31,62,61,59,55,47

Cualquier *m*-secuencia $\{a_n\}$ puede escribirse en función de las raíces del polinomio característico del LFSR mediante la función traza, de modo que $a_n = Tr(\alpha^n) = \sum_{i=0}^{L-1}\alpha^{n2^i}$. En consecuencia, dada una secuencia $\{a_n\}$ generada por un LFSR con polinomio $P_1(x)$ y raíz α, y otra secuencia $\{b_n\}$ generada por otro LFSR con polinomio $P_2(x)$ y raíz β tal que $\beta = \alpha^k$, se tiene que $a_n = \sum_{i=0}^{L-1}\alpha^{n2^i}$ y $b_n = \sum_{i=0}^{L-1}\alpha^{kn2^i}$. Esto se muestra con un ejemplo en la tabla II.

Si dos filtrados definidos por los correspondientes polinomios y funciones filtro $(P_1(x), F_1(x))$ y $(P_2(x), F_2(x))$ generan la misma secuencia, entonces se tiene que:

$$F_1(a_n, a_{n+1}, ..., a_{n+L-1}) =$$
$$= F_1(\sum_{i=0}^{L-1}\alpha^{n2^i}, \sum_{i=0}^{L-1}\alpha^{(n+1)2^i}, ..., \sum_{i=0}^{L-1}\alpha^{(n+L-1)2^i}) =$$
$$F_2(b_n, b_{n+1}, ..., b_{n+L-1}) =$$
$$= F_2(\sum_{i=0}^{L-1}\alpha^{kn2^i}, \sum_{i=0}^{L-1}\alpha^{k(n+1)2^i}, ..., \sum_{i=0}^{L-1}\alpha^{k(n+L-1)2^i}).$$

La forma algebraica normal de una función booleana permite escribir la secuencia generada por un filtrado $(P_1(x), F_1(x))$ en función de una raíz α del polinomio $P_1(x)$ y coeficientes binarios, de la siguiente manera:

$$F_1(a_n, a_{n+1}, ..., a_{n+L-1}) =$$
$$= c_0 a_n + \cdots + c_{L-1}a_{n+L-1} + c_{0,1}a_n a_{n+1} +$$
$$+ \cdots + c_{L-2,L-1}a_{n+L-2}a_{n+L-1} + \cdots +$$
$$+ c_{0,1,...L-1}a_n a_{n+1}\cdots a_{n+L-1} =$$
$$= c_0\sum_{i=0}^{L-1}\alpha^{n2^i} + \cdots + c_{L-1}\sum_{i=0}^{L-1}\alpha^{(n+L-1)2^i} +$$
$$+ c_{0,1}\sum_{i=0}^{L-1}\alpha^{n2^i}\sum_{i=0}^{L-1}\alpha^{(n+1)2^i} + \cdots +$$
$$+ c_{L-2,L-1}\sum_{i=0}^{L-1}\alpha^{(n+L-2)2^i}\sum_{i=0}^{L-1}\alpha^{(n+L-1)2^i} +$$

Tabla II
EJEMPLOS DE RELACIONES ENTRE RAÍCES, POLINOMIOS Y *m*-SECUENCIAS

Raíces	Polinomio	*m*-secuencia
$\alpha, \alpha^2, \alpha^4, \alpha^8, \alpha^{16}$	$x^5+x^4+x^3+x^2+1$	$\{a_n\}$
$\alpha^{15}, \alpha^{30}, \alpha^{29}, \alpha^{27}, \alpha^{23}$	recíproco= $x^5+x^3+x^2+x+1$	$\{b_n\}$ inverso de $\{a_n\}$
$\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24}, \alpha^{17}$	$\prod_{i=0}^{4}(x-\alpha^{3\cdot 2^i}) = x^5+x^4+x^2+x+1$	$\{c_n\}$
$(\alpha^3)^{15}=\alpha^{14}, \alpha^{28}, \alpha^{25}, \alpha^{19}, \alpha^7$	recíproco= $x^5+x^4+x^3+x+1$	$\{d_n\}$inverso de $\{c_n\}$
$\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^9, \alpha^8$	$\prod_{i=0}^{4}(x-\alpha^{5\cdot 2^i}) = x^5+x^3+1$	$\{e_n\}$
$(\alpha^5)^{15}=\alpha^{13}, \alpha^{26}, \alpha^{21}, \alpha^{11}, \alpha^{22}$	recíproco= x^5+x^2+1	$\{f_n\}$ inverso de $\{e_n\}$

$$+\cdots+c_{0,1,\ldots L-1}\sum_{i=0}^{L-1}\alpha^{n2^i}\sum_{i=0}^{L-1}\alpha^{(n+1)2^i}\cdots\sum_{i=0}^{L-1}\alpha^{(n+L-1)2^i}.$$

Por tanto, si la expresión se divide en cosets (conjuntos de enteros $E\cdot 2^i \ mod(2^L-1)$ con $0\le i\le L-1$), entonces la función se puede expresar como:

$$F_1(a_n, a_{n+1}, ..., a_{n+L-1}) =$$
$$=\sum_{i=0}^{L-1} C_{coset1}\alpha^{n\cdot coset1\cdot 2^i} + C_{coset2}\alpha^{n\cdot coset2\cdot 2^i}+\cdots$$

con $C_{cosetj}\in GF(2^L)$.

Los pesos de los cosets cuyos coeficientes son distintos de cero en la expresión anterior proporcionan información sobre el orden de la función. En particular, si se conoce la relación $\beta=\alpha^k$ entre dos filtrados $(P_1(x), F_1(x))$ y $(P_2(x), F_2(x))$ que generan la misma secuencia, entonces:

$$F_2(b_n, b_{n+1}, ..., b_{n+L-1}) =$$
$$=\sum_{i=0}^{L-1} D_{coset1}\alpha^{n\cdot coset1\cdot 2^i} + D_{coset2}\alpha^{n\cdot coset2\cdot 2^i}+\cdots$$

con $D_{cosetj}\in GF(2^L)$.

Entonces, los cosets que aparecen en ambas expresiones están vinculados de manera que para cada coset $cosetv$ en la primera expresión, existe otro coset $cosetw$ en la segunda:

$$\sum_{i=0}^{L-1} C_{cosetv}\alpha^{n\cdot cosetv\cdot 2^i} = \sum_{i=0}^{L-1} D_{cosetw}\alpha^{n\cdot cosetw\cdot 2^i}.$$

IV. FILTRADOS RECÍPROCOS

A partir de los resultados mostrados en la sección anterior, si se consideran dos LFSR con polinomios recíprocos $P_1(x)$ y $P_2(x)$, se pueden obtener dos conclusiones.

1. Si se aplica la misma función filtro $F(x)$ a ambos LFSR, se generan secuencias diferentes. Se puede utilizar el algoritmo de Berlekamp-Massey sobre las secuencias filtradas resultantes y a partir de las factorizaciones de los polinomios obtenidos, se concluye que siempre se corresponden exactamente con los mismos cosets.
2. Para generar la misma secuencia con esos LFSR, se deben utilizar dos funciones filtro diferentes $F_1(x)$ y $F_2(x)$. Dado que la factorización del polinomio obtenido con el algoritmo de Berlekamp-Massey corresponde a los cosets complementarios especulares en los grupos definidos por cada uno de los LFSR, sobre el orden de las funciones filtro influyen los pesos de esos cosets. En particular, se puede concluir que $orden(F_i) = max(L$-(peso de cada clase lateral vinculado a la factorización del polinomio de la secuencia)).

Por tanto, si existe un filtrado que produce una secuencia en la que la factorización del polinomio sólo corresponde a cosets de peso $> L/2$, entonces existe un filtrado equivalente que es menos fuerte ya que tiene orden $< L/2$. Con respecto a este filtrado equivalente, es bien sabido que el LFSR es recíproco del original. Sin embargo, descubrir cómo es la función filtro es más complejo.

Por otra parte, si una función filtro tiene orden $\sim L/2$, dado que el orden viene dado por el máximo de los pesos de los cosets asociados a la factorización, entonces existe un filtrado equivalente de orden $\ge L/2$, pues ese grado viene dado por el máximo de los pesos de los cosets. En consecuencia, si se usa un LFSR recíproco, se sabe que su peso es al menos $L-L/2$. Esto puede verse como una demostración de la conocida recomendación de uso de funciones filtro de orden $\sim L/2$.

De todo lo anterior se puede concluir que para cualquier filtrado, siempre puede obtenerse un filtrado equivalente para generar la misma secuencia, llamado filtrado recíproco. Con el fin de determinar el filtrado recíproco para cualquier filtrado conocido, el procedimiento propuesto incluye los siguientes cuatro pasos básicos:

1. Determinar las relaciones entre las raíces de los polinomios característicos del LFSR inicial y su recíproco.
2. Expresar ambas *m*-secuencias mediante la función traza.
3. Calcular los coeficientes de los cosets en la expresión de la función filtro.
4. Elegir los coeficientes adecuados para construir la función filtro recíproca.

Este procedimiento se ilustra con un ejemplo didáctico.

Ejemplo:

Dado un LFSR de longitud $L=5$, polinomio característico $P_1(x)=x^5+x^3+1$, y estado inicial $IS_1=(1,0,0,0,0)$, se aplica la función filtro de orden 4:

$$F_1(a_0, a_1, a_2, a_3, a_4) =$$
$$=a_0a_1a_3a_4+a_0a_2a_3a_4+a_0a_1a_4+a_0a_1a_3+a_1a_3a_4+$$
$$+a_0a_3a_4+a_1a_2+a_1a_3+a_2a_4+a_0a_2+a_0a_3+a_1+a_2+a_3$$

para producir la secuencia filtrada de periodo $2^5 - 1$:

$$0010110110101101110000100101011.$$

El LFSR recíproco tiene polinomio característico $P_2(x) = x^5 + x^2 + 1$, cuya raíz β se relaciona con la raíz α de $P_1(x)$ según la expresión $\beta = \alpha^{2^{5-1}-1} = \alpha^{15}$. Además, gracias al inverso modular de 15 (mod 31), puede obtenerse la relación inversa $\alpha = \beta^{29}$.

Al mismo tiempo, las *m*-secuencias $\{a_n\}$ y $\{b_n\}$ obtenidas de $P_1(x)$ y $P_2(x)$, respectivamente, pueden expresarse mediante sus expresiones traza:

$$a_n = \alpha^n + \alpha^{2n} + \alpha^{4n} + \alpha^{8n} + \alpha^{16n}$$
$$b_n = \beta^n + \beta^{2n} + \beta^{4n} + \beta^{8n} + \beta^{16n}.$$

En consecuencia, las funciones filtro F_1 y F_2 pueden expresarse en función de los $\phi(2^5 - 1)/5 = 6$ cosets $\{15, 11, 7, 5, 3, 1\}$. En particular, el coeficiente C_{15} correspondiente al coset de orden máximo 4 puede conseguirse mediante el test de presencia de raíces [16], mientras que los coeficientes C_7 y C_{11} correspondientes a los cosets de orden 3 pueden calcularse mediante agrupación de términos:

$$C_{15} = \alpha^6, C_7 = \alpha^{24}, C_{11} = \alpha^4.$$

A partir de estos valores se puede concluir que no existen más cosets de menor peso en la expresión de la función filtro F_1. Por tanto,

$$F_1 = C_{15}\alpha^{15n} + C_{15}^2\alpha^{30n} + C_{15}^4\alpha^{29n} + C_{15}^8\alpha^{27n} + C_{15}^{16}\alpha^{23n} +$$
$$+ C_7\alpha^{7n} + C_7^2\alpha^{14n} + C_7^4\alpha^{28n} + C_7^8\alpha^{25n} + C_7^{16}\alpha^{19n} +$$
$$+ C_{11}\alpha^{11n} + C_{11}^2\alpha^{22n} + C_{11}^4\alpha^{13n} + C_{11}^8\alpha^{26n} + C_{11}^{16}\alpha^{21n}.$$

Si se sustituye $\alpha = \beta^{29}$ en esa expresión, la función filtro F_2 que genera la misma secuencia se puede expresar como:

$$F_2 = C_{15}\beta^{29\cdot 15n} + C_{15}^2\beta^{29\cdot 30n} + C_{15}^4\beta^{29\cdot 29n} +$$
$$+ C_{15}^8\beta^{29\cdot 27n} + C_{15}^{16}\beta^{29\cdot 23n} + C_7\beta^{29\cdot 7n} + C_7^2\beta^{29\cdot 14n} +$$
$$+ C_7^4\beta^{29\cdot 28n} + C_7^8\beta^{29\cdot 25n} + C_7^{16}\beta^{29\cdot 19n} + C_{11}\beta^{29\cdot 11n} +$$
$$+ C_{11}^2\beta^{29\cdot 22n} + C_{11}^4\beta^{29\cdot 13n} + C_{11}^8\beta^{29\cdot 26n} + C_{11}^{16}\beta^{29\cdot 21n} =$$
$$= C_{15}\beta^n + C_{15}^2\beta^{2n} + C_{15}^4\beta^{4n} + C_{15}^8\beta^{8n} + C_{15}^{16}\beta^{16n} +$$
$$+ C_7\beta^{17n} + C_7^2\beta^{3n} + C_7^4\beta^{6n} + C_7^8\beta^{12n} + C_7^{16}\beta^{24n} +$$
$$+ C_{11}\beta^{9n} + C_{11}^2\beta^{18n} + C_{11}^4\beta^{5n} + C_{11}^8\beta^{10n} + C_{11}^{16}\beta^{20n}.$$

En consecuencia, se puede concluir que en esta expresión sólo aparecen los cosets 1, 3 y 5. Por otra parte, sus coeficientes D_1, D_3 y D_5 vienen dados por:

$$D_1 = C_{15} = \alpha^6 = \beta^{29\cdot 6} = \beta^{19}$$
$$D_3 = C_7^2 = \alpha^{24\cdot 2} = \alpha^{17} = \beta^{29\cdot 17} = \beta^{28}$$
$$D_5 = C_{11}^4 = \alpha^{4\cdot 4} = \alpha^{16} = \beta^{29\cdot 16} = \beta^{30}.$$

Dado que el peso máximo de los cosets en esa expresión es 2, se analizan los términos no lineales de orden 2 en la expresión de F_2. Como antes, para cada término no lineal de orden 2, se pueden obtener los coeficientes D_3 y D_5 mediante el test de presencia de raíces, mientras que el coeficiente D_1 correspondiente al coset de peso 1 puede calcularse agrupando términos, como se muestra en la tabla III.

Tabla III
COEFICIENTES DE LOS COSETS 3, 5 Y 1 PARA TODOS LOS POSIBLES TÉRMINOS DE ORDEN 2

	D_3	D_5	D_1
b_0b_1	β^{19}	β^{30}	β^{16}
b_0b_2	β^{7}	β^{29}	β
b_0b_3	β	β^{19}	β^{17}
b_0b_4	β^{14}	β^{27}	β^{2}
b_1b_2	β^{22}	β^{4}	β^{17}
b_1b_3	β^{10}	β^{3}	β^{2}
b_1b_4	β^{4}	β^{24}	β^{18}
b_2b_3	β^{25}	β^{9}	β^{18}
b_2b_4	β^{13}	β^{8}	β^{3}
b_3b_4	β^{28}	β^{14}	β^{19}

Una versión interesante del problema de la mochila discreta definido mediante los coeficientes de la tabla III es entonces resuelto de manera que para cada una de las dos primeras columnas correspondientes a los cosets de peso máximo, se calculan los elementos cuya suma coincide con el correspondiente coeficiente conocido. En particular, la solución muestra que los coeficientes correspondientes a los productos b_0b_2, b_1b_2, b_1b_3, b_1b_4 y b_3b_4 dan dos valores:

$$D_3 = \beta^7 + \beta^{22} + \beta^{10} + \beta^4 + \beta^{28} = \beta^{28}$$
$$D_5 = \beta^{29} + \beta^4 + \beta^3 + \beta^{24} + \beta^{14} = \beta^{30}.$$

Este resultado aplicado sobre la última columna implica que, para obtener la suma final $D_1 = \beta^{19}$, tienen que incluirse los elementos lineales $b_1 + b_2 + b_4$ en la función filtro F_2:

$$D_1 = \beta + \beta^{17} + \beta^2 + \beta^{18} + \beta^{19} + \beta + \beta^2 + \beta^4 = \beta^{19}.$$

Por tanto, se obtiene la expresión final de la función filtro equivalente:

$$F_2(b_0, b_1, b_2, b_3, b_4) =$$
$$= b_0b_2 + b_1b_2 + b_1b_3 + b_1b_4 + b_3b_4 + b_1 + b_2 + b_4.$$

Esta función aplicada sobre el LFSR recíproco con polinomio característico $P_2(x) = x^5 + x^2 + 1$ y estado inicial $IS_2 = (1, 0, 0, 1, 0)$, produce la misma secuencia que el filtrado de entrada

$$0010110110101101110000100101011.$$

F_2 es una función de orden 2 con el mismo número de términos de orden 2 y de orden 1 que F_1, pero sin términos de orden 3 ni de orden 4. Por tanto, desde un punto de vista criptográfico, el atacante podría lanzar un ataque más eficaz contra F_2 que contra F_1, aunque ambos filtrados generan exactamente la misma secuencia.

Por tanto, este ejemplo muestra que el método propuesto se puede aplicar sobre cualquier filtrado conocido para producir

un filtrado equivalente, que en el caso del LFSR recíproco es de un orden inferior. Esta es una demostración de que para algunos generadores aparentemente seguros pueden existir equivalentes más débiles, y lo que es más importante, que estos equivalentes pueden ser calculados.

V. Conclusiones y Trabajos Futuros

Este trabajo ha abordado el problema del cálculo de filtrados no lineales equivalentes que producen la misma secuencia que un filtrado conocido. En particular, se presenta el análisis de un caso en el que un LFSR recíproco se utiliza para definir un filtrado equivalente. De hecho, en esas condiciones existen relaciones específicas entre ambas funciones filtro que permiten la definición de un método específico para calcular la función filtro equivalente de una de partida. El estudio concluye que el generador equivalente puede tener un nivel de seguridad inferior al del filtrado original. Por tanto, el método propuesto permite la construcción de equivalentes más débiles que los filtrados de partida. En conclusión, este trabajo muestra que dos estructuras con niveles de seguridad aparentemente diferentes en función de sus propiedades, pueden de hecho producir exactamente la misma secuencia cifrante, por lo que en realidad ambos generadores debe ser considerados tan inseguros como el más débil de los dos.

Dada la dificultad del tema, todavía quedan muchas cuestiones abiertas. En particular, una de ellos es el desarrollo de métodos óptimos para la elección de los coeficientes correspondientes que aparecen en la última fase del método propuesto. Además, un estudio similar al que se muestra en este trabajo, pero sobre otros equivalentes que no se basen en el LFSR recíproco, podría ser útil para llevar a cabo potenciales ataques contra los filtrados no lineales.

Agradecimientos

Investigación financiada por el MINECO y la fundación Europea FEDER mediante los proyectos TIN2011-25452 e IPT-2012-0585-370000.

Referencias

[1] Biryukov, A., Shamir, A. Cryptanalytic time/memory/data tradeoffs for stream ciphers. Advances in Cryptology, ASIACRYPT00, Lecture Notes in Computer Science 1976, pp. 1-13. Springer-Verlag, 2000.

[2] Courtois, N., Klimov, A., Patarin, J., Shamir, A. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. Advances in Cryptology, EUROCRYPT00, Lecture Notes in Computer Science 1807, pp. 392-407. Springer-Verlag, 2000.

[3] eSTREAM: the ECRYPT Stream Cipher Project. Available from http://www.ecrypt.eu.org/stream/

[4] Filiol, E. Decimation attack on stream ciphers. Advances in Cryptology, INDOCRYPT 2000, Lecture Notes in Computer Science 1977, pp. 31-42. Springer-Verlag, 2000.

[5] Fuster-Sabater A., Caballero-Gil, P. On the linear complexity of nonlinearly filtered pn-sequences. Advances in Cryptology, ASIACRYPT94, Lecture Notes in Computer Science 917, pp. 80-90. Springer-Verlag, 1995.

[6] Games, R.A., Rushanan, J.J. Blind synchronization of m- sequences with even span. Advances in Cryptology, EUROCRYPT93, Lecture Notes in Computer Science 765, pp. 168-180. Springer-Verlag, 1994.

[7] Garey, M.R., Johnson, D.S. Computers and Interactability. Freeman and Company, 1979.

[8] Golic, J.D., Clark, A., Dawson, E. Generalized inversion attack on nonlinear filter generators. IEEE Transactions on Computers, 49(10), pp. 1100-1109, 2000.

[9] Golomb, S.W., Shift Register-Sequences, Aegean Park Press, Laguna Hill, 1982.

[10] Key, E.L. An analysis of the structure and complexity of nonlinear binary sequence generators. IEEE Transactions on Information Theory, 22(6), pp. 732-736, 1976.

[11] Lohlein, B. Design and analysis of cryptographic secure keystream generators for stream cipher encryption. PhD thesis, Faculty of Electrical and Information Engineering, University of Hagen, Germany, 2001.

[12] Massey, J. L. Shift-register synthesis and BCH decoding, IEEE Transactions on Information Theory, IT-15 (1), pp. 122-127, 1969.

[13] Meier W., Staffelbach, O.J. Fast correlation attacks on stream ciphers. Journal of Cryptology, 1(3), pp. 159-176, 1989.

[14] Pasalic, E. On guess and determine cryptanalysis of LFSR-based stream ciphers. IEEE Transactions on Information Theory, 55(7), pp. 3398–3406, 2009.

[15] Ronjom, S., Cid, C. Nonlinear equivalence of stream ciphers. Fast Software Encryption, pp. 40–54. Springer-Verlag, 2010.

[16] Rueppel, R.A. Analysis and Design of Stream Ciphers. Springer-Verlag, 1986.

[17] Schneider, M. Methods of generating binary pseudo-random sequences for stream cipher encryption. PhD thesis, Faculty of Electrical Engineering, University of Hagen, Germany, 1999.

[18] Siegenthaler, T. Decrypting a class of stream ciphers using ciphertext only. IEEE Transactions on Computers, 100(1), pp. 81–85, 1985.

[19] SNOW 3G specification. Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 and UIA2. Available from http://www.3gpp.org/DynaReport/35216.htm

Aportes para el estudio de anillos en ataques cíclicos al criptosistema RSA

Juan Pedro Hecht, *Universidad de Buenos Aires,* Jorge Ramió Aguirre, Abel Casado,
Universidad Politécnica de Madrid

***Resumen*—Se aporta un análisis teórico sobre un software desarrollado para analizar experimentalmente los anillos o ciclos de recifrado en el algoritmo RSA. La idea es predecir analíticamente las longitudes de anillos observadas y en casos particulares predecir la frecuencia de aparición de las distintas longitudes cuando se aplica el método a los mensajes $m \in Z_n$. También se discuten consideraciones vinculadas a la potencial factorización del módulo y la obtención de la clave privada a partir de la clave pública.**

***Palabras clave*—Aritmética modular *(modular arithmetics)*, ataque cíclico *(cycle attack)*, campos finitos *(finite fields)*, criptografía de clave pública *(public key cryptography)*, criptosistema RSA *(RSA cryptosystem)*, grupos cíclicos *(cyclic groups)*, generadores congruenciales modulares *(modular congruential generators)*, teoría de números *(number theory)*.**

I. Nomenclatura

Adoptamos la siguiente nomenclatura:

p, **q** primos
n = **pq** módulo del cifrado RSA
φ(n) = **(p-1)(q-1)** función indicador de Euler
λ(n) = **φ(n)/(p-1, q-1)** =**[p-1, q-1]**
donde **λ()** es la función de Carmichael
e coprimo con **λ(n)**
exponente público de cifrado RSA
d exponente privado de cifrado RSA
donde **ed** ≡ **1** mod **λ (n)**
m (entero, (**0** ≤ **m** ≤ **n-1**) mensaje a cifrar
c (cifrado) = $\mathbf{m^e}$ (mod **n**)
k longitud de anillo o período de ciclo
$\mathbf{Z_n}$ enteros módulo n
$\mathbf{Z_n^*}$ enteros módulo n coprimos con n
a|**b** a divide a b
(a, b) máximo común divisor de a y b
[a,b] mínimo común múltiplo de a y b
{**A**}conjunto A
|**A**|cardinal u orden del conjunto A
$<g>$ grupo cíclico generado por g
$\mathbf{ord_m}\,(\mathbf{e})$ orden multiplicativo de e módulo m

II. Introducción

Fruto de la investigación realizada durante los últimos meses de 2013 e inicio de 2014 en el Proyecto Fin de Grado de título "Software para el estudio del comportamiento de los ataques por cifrado cíclico en RSA" [1] en la Escuela Técnica Superior de Ingeniería de Sistemas Informáticos de la Universidad Politécnica de Madrid, su director presenta en [2] unos primeros resultados sobre la generación de anillos que se producen en dicho ataque, utilizando el software educacional RingRSA desarrollado en el proyecto que será próximamente de dominio público, planteándose un interesante tema algebraico vinculado a dos aplicaciones criptográficas; a saber, el orden de los generadores congruenciales modulares (ver Blum-Blum-Shub [3]) y el ataque cíclico al criptosistema RSA [3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13]. Ambos han sido estudiados de manera extensa a lo largo de los años; basta consultar la bibliografía citada en las referencias arriba indicadas.

Para el caso del ataque a RSA, está demostrado que este método de cifrado cíclico permite descifrar un criptograma que cuenta con confidencialidad, es decir un mensaje secreto que se ha cifrado con la clave pública de destino, contando para ello solamente con los datos públicos de la clave en cuestión de la víctima, esto es los valores n y e. En teoría, permite además quebrar el criptosistema ya que brinda un método para obtener la clave privada d a partir de la clave pública e a través de la factorización del módulo [3, 4]. El objetivo del presente trabajo es:

a. Computar analíticamente las posibles longitudes de los anillos generados por el programa RingRSA.

b. Para determinados casos, calcular analíticamente las frecuencias de aparición de las distintas longitudes de los anillos generados por el programa RingRSA.

Como se demuestra más adelante, ambos objetivos se cumplen satisfactoriamente.

III. Planteamiento del problema

El problema central consiste en computar sucesivamente cifrados iterados (mod n) hasta completar un ciclo u órbita del exponente.

$$m^e,\ m^{e^2},\ m^{e^3}, ...,\ m^{e^k} \equiv m\ \ (mod\ n)$$

$$m\ \in Z_n\ y\ \ (e, \lambda\,(n)) = 1$$

Dado que las referencias a las ecuaciones serán siempre locales, éstas no se enumerarán.

La existencia de k está asegurada [3, 9, 12]; existe un exponente de cifrado para el cual se recupera el mensaje m, exactamente una iteración previa al volver al cifrado original c [3].

Queda claro que los sucesivos exponentes de recifrado forman un grupo multiplicativo cíclico cuyo generador es el exponente e:

$$< e > \ : \left\{e^1,\ e^2,\ e^3,\ e^4, ..., e^k \equiv 1\right\}\ \ mod\ \lambda\,(n)$$

$$k = ord_{\lambda(n)}(e)$$

Donde k es el orden multiplicativo de e módulo $\lambda(n)$. El máximo valor de k, la cota superior de recifrados necesarios para recuperar el mensaje, está dada por:

$$\left|Z^*_{\lambda(n)}\right| = max(k) \;=\; \lambda(\lambda(n))$$

Los posibles valores de k son divisores de max (k) [12]:

$$k \mid \lambda(\lambda(n))$$

ya que los órdenes de los subgrupos son divisores del orden de un grupo [10].

IV. Distribución estadística de valores $Z^*_{\lambda(n)}$

Para el estudio de la distribución estadística, se buscan primos p, q al azar y sin estructura.

Sabemos que los posibles valores de k son divisores de $\lambda(\lambda(n))$. Se puede estudiar su distribución analítica o estadísticamente según cómo se elijan los valores de los primos p, q.

Desde el punto de vista estadístico, si se define r el mayor factor primo del indicador (y de lambda) de uno de los primos, digamos p, entonces r es un divisor de p-1. Acorde a lo demostrado en [9] respecto a la factorización típica y el mayor factor primo, r tendrá un tamaño cercano al 63 % del tamaño de p-1. O sea para n de 1.024 bits, r será del orden de 322 bits (el 63 % de 512 bits). Según [4] esto implica que:

1. *Si se elige al azar un e cuyo k es múltiplo de r, la longitud del ciclo será al menos de tamaño r, lo que hace inviable el ataque cíclico.*
2. *La probabilidad de elegir al azar un e con un orden multiplicativo que no sea múltiplo de r es 1/r, lo que resulta despreciable (para n de 1024 bits p˜10^{-322}). La deducción de esta consecuencia puede obtenerse de la discusión planteada en [4].*
3. *Debe quedar en claro que lo antedicho es válido para todo valor e.*

En conclusión, la inmensa mayoría de pares de los valores (m,e,p) elegidos para el ataque cíclico resultan ser extremadamente costosos en tiempo [4]. Obviamente, los ensayos computacionales con RingRSA así lo ilustran. Dado que el generador congruencial modular representa un generador pseudoaleatorio de calidad criptográfica [3] y que no parece haber regularidad en la distribución de divisores de $\lambda(n)$, no tiene sentido buscar una función de densidad de probabilidades a priori cuando los primos se eligen al azar y son desconocidos [7]. Sin embargo, para ciertos valores especiales de primos p, q es posible dar una respuesta. Para dar sustento analítico a lo antedicho, en el siguiente apartado se deduce que es posible obtener la función de densidad de longitudes de anillos si se conoce la factorización del módulo n en sus primos componentes.

V. Distribución analítica de longitudes de ciclos en $Z^*_{\lambda(n)}$

Formalizando el análisis de los generadores congruenciales modulares, se parte de

$$m_y \equiv {m_{y-1}}^e \,(mod\ n) \qquad 0 \le m_y \le n-1$$

$$y = 1, 2, 3, ...$$

Donde m_0 el valor inicial debe ser coprimo con n y el exponente $e \ge 2$. Se demuestra [6] que si $(e,\ \lambda(n)) = 1$ la secuencia resulta periódica pura (sin fase aperiódica inicial) con período k, entonces

$$k = ord_x(e) \;\; donde\ x = ord_n(m_0)$$

Evidentemente esta última relación [6] permite calcular, para cada m_0 coprimo con n y cada e coprimo con $\lambda(n)$, la longitud del período multiplicativo o longitud del anillo, con lo cual se cumple uno de los objetivos del trabajo. Obviamente cada resultado obtenido con RingRSA así lo verifica.

De aquí en adelante consideramos que m y e se ajustan a las condiciones que generan secuencias periódicas puras. Para estudiar la distribución analítica, se buscan primos p, q con estructura especial.

Sea n = pq donde p, q son primos de Sophie Germain. Entonces p = 2r+1 y q = 2s+1 donde r, s son primos, que si también fuesen de Sophie Germain, r = 2t+1 y s = 2u+1 donde t, u son primos. Ambos conjuntos de primos (t, r, p) y (u, s, q) constituyen cadenas de Cunningham de primera especie.

Los órdenes multiplicativos posibles de los integrantes del grupo multiplicativo Z*$_n$ de cifrados RSA [9, 10] son los divisores de:

$$\lambda(n) = \frac{\varphi(n)}{MCD(p-1, q-1)} = \frac{(p-1)(q-1)}{2} = 2rs$$

De las propiedades bien conocidas de los grupos cíclicos (por ejemplo ver 2.172 y 2.173 de [3]) surge claramente que cada divisor d | 2rs define un único subgrupo cíclico de orden φ(d). Obviamente se cumple el conocido teorema [10]

$$\varphi(n) = \sum_{d|\varphi(n)} \varphi(d)$$

y el cardinal del grupo es:

$$|Z^*_n| = \varphi(n) = 4rs$$

Así resulta que φ(d) es la multiplicidad de veces con las cuales se presenta cada divisor d, o sea cuántos elementos tienen un orden determinado. A su vez, los órdenes multiplicativos de los integrantes del grupo multiplicativo $Z^*_{\lambda(n)}$ de exponentes de cifrados RSA [5, 6, 7, 8] son los divisores de:

$$\lambda(\lambda(n)) = [\lambda(2), \lambda(r), \lambda(s)] = [1, 2t, 2u] = 2tu$$

Y el cardinal del grupo en cuestión es [12]:

$$|Z^*_{\lambda(n)}| = \varphi(\lambda(n)) = \varphi(2)\,\varphi(r)\,\varphi(s) =$$

$$= 1\,(2t)\,(2u) = 4tu$$

También los exponentes forman un grupo multiplicativo cíclico, por lo tanto vale lo expresado antes (2.172 y 2.173 de [3]), por lo cual es evidente que cada divisor d $\mid \lambda(\lambda(n))$ define un único subgrupo cíclico de orden φ(d) y se cumple [10]:

$$\lambda(\lambda(n)) = \sum_{d|\lambda(\lambda(n))} \varphi(d)$$

Para $\lambda(n) = 2rs$, resulta:

$$c \in \ Z^*_{\lambda(n)},\ donde\ c \equiv a\ \ (mod\ r)\ \ y\ \ \ c \equiv b\ \ (mod\ s)$$

y se cumple la siguiente relación entre los órdenes de ciclos de potencias modulares:

$$ord\,(c,\ \lambda(n)) = [ord\,(a,\ r),\ ord\,(b,s)]$$

Y se generan los siguientes subgrupos de exponentes para Z^*_r y Z^*_s:

Cuadro I: SUBGRUPOS DE EXPONENTES PARA Z^*_r

Orden	Multiplicidad	Detalle
1	1	=$\varphi(1)$
2	1	=$\varphi(2)$
t	t-1	=$\varphi(t)$
2t	t-1	=$\varphi(2t)$
Suma	2t	=$\lvert Z^*_r \rvert$

Cuadro II: SUBGRUPOS DE EXPONENTES PARA Z^*_s

Orden	Multiplicidad	Detalle
1	1	=$\varphi(1)$
2	1	=$\varphi(2)$
u	u-1	=$\varphi(u)$
2u	u-1	=$\varphi(2u)$
Suma	2u	=$\lvert Z^*_s \rvert$

Cuando se computan las multiplicidades de los órdenes en $Z^*_{\lambda\ (n)}$ hay que considerar las combinaciones de órdenes de Z^*_r y Z^*_s .

Por ejemplo, para el orden 2 en $Z^*_{\lambda\ (n)}$ es necesario que la combinación de órdenes en Z^*_r y Z^*_s genere un mcm cuyo resultado sea 2. En este caso, hay tres combinaciones posibles con ese resultado : [1,2], [2,1] y [2,2]. La multiplicidad de 2 en $Z^*_{\lambda\ (n)}$ resulta en la suma de los productos de las multiplicidades de estas tres combinaciones:

$$Multiplicidad\ \left(orden\ 2\ \ en\ \ Z^*_{\lambda(n)}\right)$$

$$= \varphi(1)\,\varphi(2) + \varphi(2)\,\varphi(1) + \varphi(2)\,\varphi(2) = 3$$

Finalmente, los divisores de $\lambda(\lambda(n)) = 2tu$ resultan ser:

$$\{1,\ 2, t,\ u,\ 2t,\ 2u,\ tu,\ 2tu\}$$

y generan los siguientes subgrupos de exponentes que se presentan en la siguiente tabla.

Cuadro III: SUBGRUPOS DE EXPONENTES EN $Z^*_{\lambda(n)}$

Orden	Multiplicidad	Detalle
1	1	=$\varphi(1)\varphi(1)$
2	3	=$\varphi(1)\varphi(2) + \varphi(2)\varphi(1) + \varphi(2)\varphi(2)$
t	t-1	=$\varphi(t)$
u	u-1	=$\varphi(u)$
2t	3t-3	=$\varphi(2t)\varphi(1) + \varphi(t)\varphi(2) + \varphi(2t)\varphi(2)$
2u	3u-3	=$\varphi(1)\varphi(2u) + \varphi(2)\varphi(u) + \varphi(2)\varphi(2u)$
tu	tu-t-u+1	=$\varphi(t)\varphi(u)$
2tu	3tu-3t-3u+3	=$\varphi(2t)\varphi(u) + \varphi(t)\varphi(2u) + \varphi(2t)\varphi(2u)$
Suma	4tu	=$\lvert Z^*_{\lambda(n)} \rvert$

Disponiendo de esta tabla y su segunda columna, verdadero aporte de este trabajo al no existir antecedentes publicados, se posee la distribución estadística de las frecuencias de aparición de los ocho órdenes posibles, cualesquiera sean los primos t, u y el valor del mensaje m y exponente e al aplicar el ataque cíclico al RSA. A continuación se ilustra su aplicación a un ejemplo numérico sobre dos cadenas de Cunningham de primera especie.

VI. Ejemplo numérico de la distribución analítica de valores de longitudes de ciclos en $Z^*_{\lambda(n)}$

Adoptando los símbolos del apartado previo, sean:

t = 5, r = 11, p = 23 y u = 89, s = 179, q = 359 cadenas de primos Cunningham de primera especie. Así resulta:

n = pq = 8.257

$|Z^*_n| = \varphi(n) = (p-1)(q-1) = (2r)(2s) = 4rs = 7.876$

$\lambda(n) = \varphi(n)/2 = 2rs = 3.938$

$|Z^*_{\lambda(n)}| = \varphi(\lambda(n)) = \varphi(2rs) = \varphi(2)\varphi(r)\varphi(s)$
$= 1*2t*2u = 4tu = 1.780$

$\lambda(\lambda(n)) = [\lambda(2), \lambda(r), \lambda(s)] = [1, 2t, 2u] = 2tu = 890$
Divisores de λ (λ(n)) = {1, 2, t, u, 2t, 2u, tu, 2tu} = {1, 2, 5, 89, 10, 178, 445, 890]

Y se generan los siguientes ciclos de anillos:

Cuadro IV: SUBGRUPOS DE EXPONENTES EN $Z^*_{\lambda(n)}$

Orden	Multiplicidad	Detalle del cómputo
1	1	$=1$
2	3	$=3$
5	4	$=t-1$
89	88	$=u-1$
10	12	$=3t-3$
178	264	$=3u-3$
445	352	$=tu-t-u+1$
890	1056	$=3tu-3t-3u+3$
Suma	1780	$=\|Z^*_{\lambda(n)}\|$

Es menester aclarar que esta tabla emplea las mismas fórmulas deducidas para la tabla final del apartado previo. Cualquiera sea m^e (mod n), la longitud del ciclo de recifrado será un miembro de la lista de la primera columna. La función de densidad de longitudes se desprende de la segunda columna. Analizando dicha columna, se observa que la mayoría de longitudes de ciclo son relativamente grandes. Así de las ocho longitudes de ciclo disponibles para combinaciones de (m, e) tomadas al azar, el 60 % posee la máxima longitud posible.

En la figura 1 se genera esta clave con RingRSA eligiendo un exponente válido al azar.

Una vez generada la clave, puede comprobarse de manera experimental que las longitudes de los ciclos de recifrado son miembros divisores de $\lambda(\lambda(n))$, que están reflejados en la lista de la primera columna.

Este exponente, al igual que cualquier otro, genera su órbita de valores divisores. En la Figura 2 se ven que están presentes los valores de la primera columna {1, 10, 89, 890}; pero no los valores {2, 5, 178, 445}.

Por ejemplo si ahora elegimos e=31, se generan anillos de longitudes {1, 5, 89, 445} como se nuestra en la figura 3.

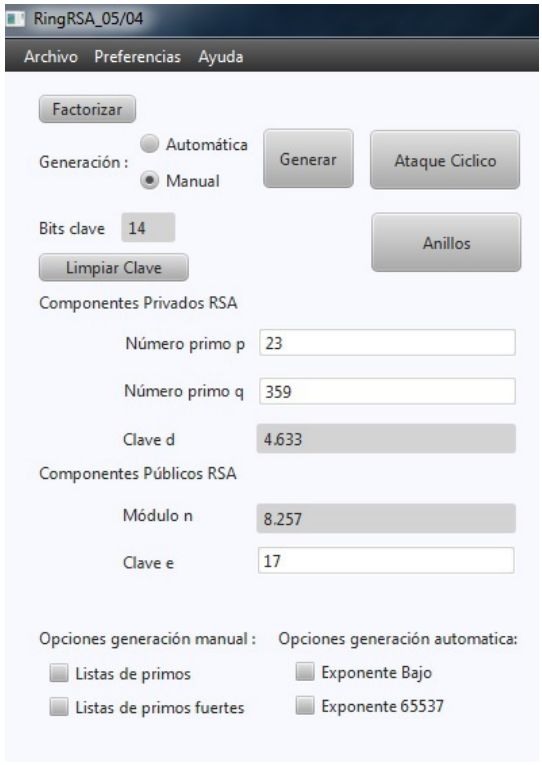


Figura 1: Generación de la clave p=23, q=359, e=17

VII. Trabajos Futuros

1. *El ataque cíclico al RSA como algoritmo de factorización*

   Si se obtiene alguna longitud de anillo (orden k de recifrado), parece ser posible llegar a factorizar [4] y obviamente obtener luego la clave privada. Este tema está siendo analizado por nosotros. También hay evidencia de que es posible factorizar si se obtiene computacionalmente un múltiplo de φ(n) [13]. De todas maneras, cabe añadir que completar experimentalmente un anillo no es tarea fácil si el orden del módulo es suficientemente grande, por lo cual el método de factorización de ataque cíclico es inviable en la práctica. Sin embargo, el ataque siempre es teóricamente exitoso al menos para obtener el mensaje original.

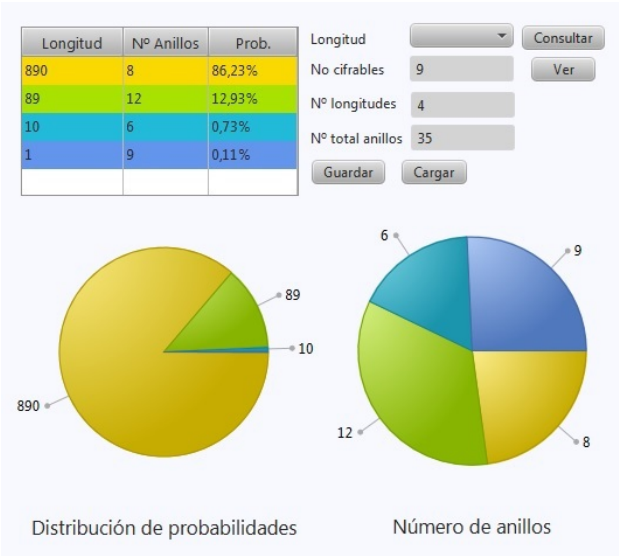


Figura 2: Anillos que se generan en la clave p = 23, q = 359 utilizando e = 17

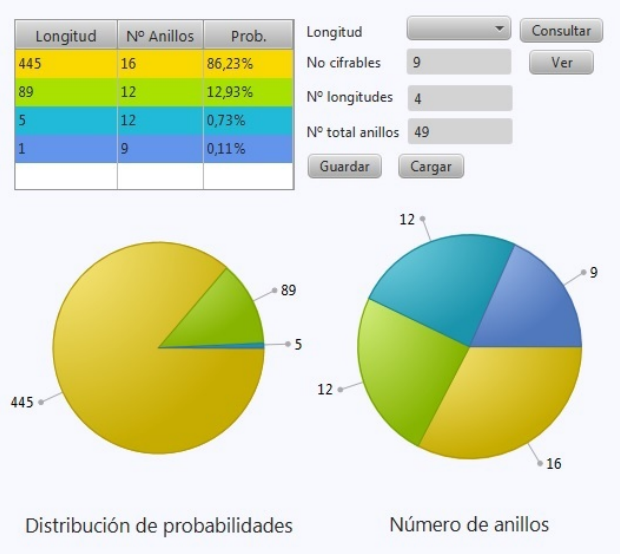


Figura 3: Anillos que se generan en la clave p = 23, q = 359 utilizando e = 31

2. *Cómo prevenir el ataque cíclico al RSA*

Para prevenir o dificultar que prospere el quiebre de RSA a través de un ataque cíclico [12] se debe cumplir, dados los recursos actuales, que:

$$\lambda(\lambda(n)) \quad y \quad ord_{\lambda(n)}(e) > 10^{200}$$

Esta condición se logra con primos fuertes p, q tales que:

$$\frac{p-1}{2} \quad y \quad \frac{p-3}{4} \; sean \; primos$$

Y lo mismo para q. Si p y q lo fuesen, se verifica:

$$\lambda(n) = 2\frac{p-1}{2}\frac{q-1}{2}$$

$$\lambda(\lambda(n)) = mcm\left(2, \frac{p-3}{2}, \frac{q-3}{2}\right) = (p-3)(q-3)/8$$

Sin embargo hay opinión fundada que la primera condición $\frac{p-1}{2}$ sería suficiente para bloquear el ataque cíclico [3].

3. *Complejidad computacional del ataque cíclico*

Este ataque, a pesar de ser teóricamente eficaz, parece pertenecer a la clase de complejidad NP, es decir un problema de tanta dificultad como el propio problema de la factorización de enteros [3, 12]. La evidencia empírica de lo antedicho es que si n se incrementa lo suficiente, el ataque cíclico se vuelve inútil con los recursos computacionales actuales. De hecho, en [6] se demuestra que si se consideran primos tomados al azar, el ataque cíclico es de complejidad $O(\sqrt{n})$, lo que lo hace equivalente al método de factorización por división de primos menores a la raíz cuadrada [9].

No obstante, la aparición de anillos basados en el cifrado cíclico y comentado en este artículo, presenta interesantes aspectos para un posterior estudio. A modo de ejemplo, una clave RSA de 50 bits con primos p y q de 25 bits cada uno (n = pq = 29.221.417 x 24.917.353 = 728.120.362.549.201) para e=15.131 encuentra el mensaje secreto 2 en 300 milisegundos, realizando tan sólo 259.956 cálculos; pero si la clave pública es e=65.537, el mismo ataque tardará 21 segundos al necesitar ahora realizar 13.777.668 cálculos.

En la figura 4 se muestra el resultado del ataque cíclico al mensaje 2 para dicha clave con exponente 65.537 en 21 segundos.

Más aún. Con RingRSA elegimos otra clave de dimensiones similares pero con un tamaño un 20 % menor (n = pq = 20.713.829 x 28.164.079 = 583.385.916.348.491) usando e=65.537 y se buscan los anillos de la clave comenzando como siempre por el número 2. Después de mil minutos, se detiene la búsqueda y con 48.040.766.272 valores ya obtenidos no se encuentra aún ese primer anillo. Queda por tanto la posibilidad de que el número 2 se encuentre en uno de estos cuatro divisores de Carmichael: 54.452.402.235, 72.603.202.980, 108.904.804.470 o bien 217.809.608.940 [14]. Para el mayor divisor de Carmichael de esta clave, a una tasa de 800.000 cifrados por segundo, el programa tardaría más de 75 horas en encontrar ese anillo.

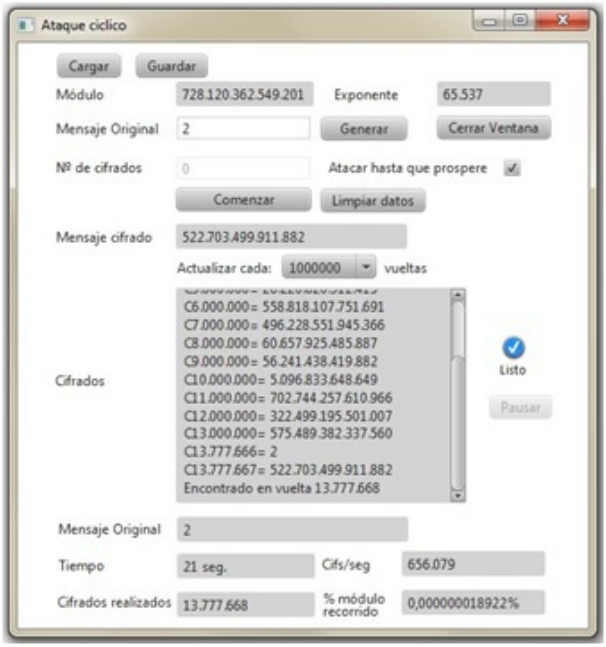


Figura 4: Ataque al valor secreto 2 para la clave con datos públicos n = 728.120.362.549.201 y e = 65.537

En otras palabras, para un par de claves muy similares, en el primer caso el ataque por cifrado cíclico requerirá un cómputo y un tiempo casi un millón de veces menor que en el segundo, siendo incluso la primera clave un 20 % mayor que la segunda. Otro aspecto interesante a estudiar en el futuro.

VIII. Conclusiones

Se ilustra que es factible calcular las longitudes de ciclos (anillos), las frecuencias de longitudes de ciclos de recifrado en el método de ataque cíclico al RSA para elecciones determinadas de valores n, m, e y que a partir de un único ataque cíclico exitoso al RSA sería posible, en teoría, hallar la clave privada a partir de la pública.

Este trabajo plantea algunos temas abiertos como trabajo futuro, entre los que se encuentran los presentados en el apartado anterior: la deducción práctica de la clave privada d mediante este tipo de ataques cíclicos completando un solo anillo, la posibilidad de dificultar o bloquear este ataque si se utilizan primos seguros en el diseño de las claves, todo ello con el soporte del mencionado software y el correspondiente desarrollo analítico.

Referencias

[1] A. Casado, J Ramió, Software para el estudio del comportamiento de los ataques por cifrado cíclico en RSA, programa RingRSA de próxima publicación en Internet, Proyecto Fin de Grado de Abel Casado Gimeno, ETSISI-UPM, 2014.

[2] J Ramió, , RSA cumple 36 años y se le ha caducado el carné joven, Rooted CON Madrid,Marzo 2014. http://www.criptored.upm.es/guiateoria/gt_m001k1.htm

[3] A. J. Menezes, P. C. van Oorschot and S. A. Vanstone, Handbook of Applied Cryptography, CRC Press, Boca Raton, FL, 1996.

[4] T. Pornin, Cycle Attack on RSA, http://crypto.stackexchange.com/questions/1572/cycle-attack-on-rsa

[5] T. W. Cusick, Properties of the x2 mod N pseudorandom number generator, IEEE Trans. Inform. Theory, 41 (1995), 1155?1159.

[6] J. B. Friedlander, C. Pomerance, and I. E. Shparlinski Period of the power generator and small values of Carmichael's function, Math. Comp., 70 (2001), 1591.1605.Corrigendum.Math.Comp., 71 (2002), 1803.1806.

[7] M. Gysin and J. Seberrye, Generalized cycling attacks on RSA and strong RSA primes, Chapter 3, J. Pieprzyk, R. Safavi-Naini, and J. Seberry (Eds.): ACISP'99, LNCS 1587, pp. 149-163, 1999. Springer-Verlag Berlin Heidelberg, 1999.

[8] H. Riesel, Prime Numbers and Computer Methods for Factorization, Progress in Mathematics, 2nd Ed., Birkhäuser, Boston, 1994.

[9] R. Lidl and H. Niederreiter, Finite Fields, Cambridge University Press, Cambridge, 1997.

[10] N. J. A. Sloane, Sequence A181776 a(n) = lambda(lambda(n)), where lambda(n) is the Carmichael lambda function, The On-Line Encyclopedia of Integer Sequences, http://oeis.org/A181776

[11] G. L. Miller Riemann's hypothesis and tests for primality, Journal of Computer and System Sciences, 13:3, 300-317, 1976.

[12] Divisores de Carmichael para el módulo n=583.385.916.348.491. https://www.wolframalpha.com/input/?i=Divisors[Carmichael[CarmichaelLambda[583385916348491]]]

Pedro Hecht (M 2012) nació en Buenos Aires, Argentina, el 14 de Julio de 1944. Se graduó como Licenciado en Análisis de Sistemas (ESIO-DIGID) y como Doctor de la Universidad de Buenos Aires. Actualmente es Profesor Titular de Criptografía I y II de la Maestría en Seguridad Informática dependiente de las Facultades de Cs. Económicas, Cs. Exactas y Naturales y de Ingeniería de la Universidad de Buenos Aires (UBA) e idéntico cargo en la Facultad de Ingeniería del Ejército (EST). Además es el Coordinador Académico de la citada Maestría (UBA) y es investigador en modelos matemáticos en UBACyT (UBA). Es miembro de Criptored, IEEE Argentina, ACM SIGCSE, ACM SIGITE y otras. Áreas de interés actual: álgebra no conmutativa aplicada a la criptografía.

Jorge Ramió Jorge Ramió nació en Barcelona, España, el 22 de Enero de 1952. Doctor en Sistemas Inteligentes en la Ingeniería por la Universidad de León (2013), Máster en Ingeniería de Sistemas y Servicios Accesibles para la Sociedad de la Información por la Universidad Politécnica de Madrid (2011) y Doctor Ingeniero de Telecomunicación Diplomado por la Universidad Politécnica de Madrid (1983). Creador de Criptored y sus proyectos derivados intypedia y crypt4you, así como el Congreso Iberoamericano CIBSI y el Taller de Enseñanza TIBETS. Imparte asignaturas de seguridad y criptografía en la UPM desde el año 1994.

Profesor invitado en posgrados de España y Latinoamérica, actualmente se encuentra desarrollando los proyectos de píldoras formativas Thoth y Mapa de Enseñanza de la Seguridad de la Información MESI.

Abel Casado nació en Madrid, España, el 2 de Noviembre de 1989. Está finalizando los estudios de Grado de Ingeniera de Computadores en la Escuela Técnica Superior de Ingeniería de Sistemas Informáticos, (ETSISI-UPM), a falta de entregar el PFG. Actualmente se encuentra trabajando como becario para el Departamento de Matemática Aplicada de la ETSISI, llevando a cabo labores relativas a geometría computacional y renderizado en 3D en la empresa ACA España. Áreas de interés actual: Computación Gráfica, Desarrollo de aplicaciones Java, Criptografía y Seguridad de la Información.

Modelado de un criptoprocesador mediante LISA

José Molins
Departamento de Ingeniería
Mecanizados Escribano S.L.
Email: jose.molins@mecaes.es

Rafael Rico
Departamento de Automática
Universidad de Alcalá
Email: rafael.rico@uah.es

***Resumen*—Para acortar los ciclos de diseño y desarrollo de criptoprocesadores se propone utilizar LISA, un lenguaje de descripción de arquitecturas programables. LISA, además, permite minimizar el coste hardware del procesador criptográfico así como su consumo. Como caso de estudio se propone un operador que da soporte al código operación de la exponenciación modular. Desde el punto de vista del hardware, el operador de exponenciación se implementa mediante cauces multifunción aptos para el procesamiento vectorial. Como característica más sobresaliente, las operaciones se podrán efectuar sobre operandos de tamaño arbitrario, es decir, la longitud k de la representación binaria de los operandos no estará limitada por el tamaño n de los registros del operador y se pasará como un operando más del código de operación.**

***Palabras clave*—ASIP, Criptografía (Cryptography), Criptoprocesador (Cryptoprocessor), Exponenciación modular (Modular exponentiation), RSA, M2M, Montgomery, LISA.**

I. Introducción

En los últimos años, el mercado de la conectividad máquina a máquina (M2M) ha aumentado notablemente. La necesidad de incluir mayor seguridad en los sistemas crece día a día. Se prevé que a nivel global la demanda de dispositivos con seguridad añadida se eleve hasta los 529 millones de unidades en 2017 [1] especialmente en sistemas empotrados y en sistemas que gestionan el Internet de las Cosas (IoT).

Los criptosistemas están diseñados para proporcionar los servicios de confidencialidad, autenticación e integridad de datos. En este escenario los procesadores criptográficos se están haciendo imprescindibles.

En este trabajo se propone LISA como herramienta de diseño. Su uso favorece minimizar el tiempo de desarrollo, el coste hardware y el consumo energético. LISA es un lenguaje de descripción de arquitecturas programables y repertorios de instrucciones, unificado en una sola herramienta y centrado en facilitar el diseño y desarrollo de procesadores de propósito específico como son los criptoprocesadores. Los requisitos del procesador se describen sobre 6 modelos arquitecturales. Como caso de estudio se propone un ejemplo de instrucción de un criptoprocesador implementada sobre un operador que calcula la exponenciación modular mediante Montgomery.

La arquitectura del operador está basada en 3 cauces multifunción configurables. El código de operación ejemplo utiliza dichos cauces para computar el algoritmo de Montgomery sobre una longitud de operandos de k-bits parametrizable. Evidentemente, los 3 cauces multifunción son capaces de realizar otras muchas operaciones (sumas, desplazamientos, cálculos modulares, etc.) asociadas a otros tantos códigos de operación del repertorio de instrucciones. Esta característica contribuye a la minimización del coste hardware y del consumo.

El artículo está estructurado de la siguiente manera: En la Sección II se hace una somera revisión del estado del arte de los criptosistemas. En la Sección III se describe LISA mostrando el flujo de trabajo y las mejoras que aporta al desarrollo de criptoprocesadores. La Sección IV muestra un caso de estudio referente a la implementación de un código de operación para efectuar la exponenciación modular mediante un operador implementado con cauces multifunción. Y ya por último, en la Sección V se ofrecen algunas conclusiones.

II. Trabajos relacionados

Los distintos sistemas hardware destinados a dar servicios criptográficos se diseñan en base a primitivas, algoritmos y protocolos criptográficos. Éstos pueden clasificarse en distintos tipos en función de las necesidades: procesadores de propósito general modificados, criptocoprocesadores, criptoprocesadores y criptoarrays.

Los procesadores de propósito general (PPG) ofrecen una gran flexibilidad pero no son apropiados para aplicaciones de alto rendimiento. Las amenazas de seguridad pueden aparecer asociadas a cualquiera de los niveles de abstracción del dispositivo.

Los procesadores de propósito general modificados contienen unidades funcionales para ejecutar operaciones criptográficas tales como Data Encryption Standard (DES) o Advanced Encryption Standard (AES). Estas unidades están asociadas a un conjunto de instrucciones específicas. Esta solución aumenta el rendimiento aunque el diseño del repertorio de instrucciones adicional presenta una gran dificultad. Un ejemplo claro de ello es el procesador CryptoBlaze [3] basado en el soft-core de Xilinx Picoblaze. Cryptoblaze ofrece una serie de funciones y códigos de operación enfocados a funciones criptográficas. Recientemente los procesadores Intel [4] comercializados a partir de 2010 incluyen en sus repertorios 6 nuevas instrucciones específicas para trabajar con un operador criptográfico AES implementado en la arquitectura del procesador.

El término criptocoprocesador hace referencia a un dispositivo hardware dedicado a la ejecución de funciones criptográficas. Estos dispositivos no son programables pero puede ser controlados y configurados utilizando un procesador host. Un ejemplo de este concepto es la arquitectura SAFES [5]

que propone un sistema que utiliza monitores para detectar distintos tipos de ataques. Así, la contra-medida del sistema de seguridad depende del ataque perpetrado. Pueden existir una contra-medida individual o una contra-medida global cuando se intenta una modificación del sistema. Esta arquitectura posee una parte reconfigurable en lógica programable donde se implementan las primitivas de seguridad. Por el contrario, los criptoprocesadores son dispositivos hardware programables con un conjunto de instrucciones especializado en funciones criptográficas que incorporan uno o varios operadores dedicados a la computación criptográfica.

El procesador Criptonita [6] posee un repertorio específico enfocado a operaciones tales como permutación, rotación y XOR. La propuesta HCrypt [7] es segura ante ataques software ya que la ruta de datos se divide en dos zonas: una zona protegida (claves y su almacenamiento) y una zona no protegida (almacenamiento de datos y proceso). Las claves almacenadas nunca pueden pasar sin cifrar a los registros de datos. Los criptoprocesadores están diseñados para ser empotrados en entornos multiprocesador system-on-chip (MP-SoC) o en procesadores dedicados (como los DSP) representando una eficiente solución en términos de flexibilidad, prestaciones y rendimiento. Los sistemas multi-núcleo aumentan el rendimiento y proporcionan soluciones para aplicaciones con necesidades de cálculos criptográficos heterogéneos siendo posible la configuración al vuelo en lógica programable.

Otro tipo de dispositivo seguro es el criptoarray. La investigación ha demostrado que las arquitecturas reconfigurables son altamente eficientes para computación intensiva en sistemas empotrados y para seguridad en los datos. La arquitectura Celator [8] es un ejemplo de ello. Está compuesta de una matriz de elementos de proceso optimizada para computar paralelamente el algoritmo AES y otros cifradores simétricos como DES. El encaminamiento de las unidades de proceso de Celator es configurable, siendo la lógica de control la que determina la arquitectura y los elementos de proceso.

Los procesadores criptográficos citados trabajan con operadores de un tamaño k fijo, es decir, con longitudes de 64, 512 o 1024 bits como máximo.

III. Flujo de diseño LISA

Actualmente las técnicas de diseño de procesadores de propósito especifico (ASIP) tales como los criptoprocesadores, implican un diseño manual de las herramientas de desarrollo. El flujo de diseño de los ASIP cuenta primariamente con un diseño de la arquitectura, un diseño de las herramientas software (compilador, enlazador, etc.) y, por último, una fase de integración y validación del conjunto. Esta metodología conlleva una separación de los grupos de trabajo de ingeniería y fácilmente supone aumentar el tiempo del desarrollo en el caso que uno de los grupos se demore en sus tareas.

En este artículo se propone LISA (Language for Instruction-Set Architecture) como un lenguaje, una metodología y un entorno de trabajo adecuado para desarrollar un criptoprocesador de manera eficiente. LISA es, principalmente, un lenguaje enfocado a describir repertorios de instrucciones y comportamientos de arquitecturas ASIP. La estructura del lenguaje esta basada en la declaración de recursos y de las operaciones que se van a realizar. El flujo de trabajo de LISA se desarrolla en cuatro fases principales [9]. Primera: exploración de la arquitectura. Se realiza el co-diseño del sistema decidiendo qué tareas se harán por hardware y cuáles se harán por software. También se lleva a cabo el diseño del repertorio de instrucciones. Por último se diseña la micro-arquitectura que dará soporte a ese repertorio. Segunda: implementación de la arquitectura. Se genera un modelo HDL del ASIP donde, con herramientas de síntesis y a nivel de puertas, se analiza el tamaño ocupado en silicio, velocidad del sistema y consumo de potencia. Si algunos de estos parámetros no son los esperados será necesario iterar de nuevo sobre la primera fase. Tercera: generación de las herramientas software para el desarrollo de aplicaciones. Es prácticamente automática dando lugar al compilador-C, linkador y debugger para la arquitectura diseñada. Por último, la cuarta: integración y verificación. Se generan los simuladores para verificar el correcto funcionamiento de las herramientas software con la arquitectura hardware.

Para que LISA pueda generar arquitectura y herramientas de desarrollo son necesarios una serie de requisitos agrupados en 6 modelos [9]. En primer lugar, un modelo de memoria donde se definen los registros de la arquitectura y los rangos de la memoria de datos y de la memoria de programa. En el segundo modelo se describen los recursos requeridos y unidades funcionales utilizados por las operaciones que se van a ofrecer. El tercer modelo describe el comportamiento de las estructuras del hardware y se utiliza únicamente para fines de simulación. El cuarto modelo, uno de los más importantes, describe el repertorio de instrucciones, expresando la sintaxis, la codificación y los operandos de una manera muy rápida e intuitiva. El quinto, es el modelo temporal que especifica la activación secuencial de los distintas operaciones en el hardware. Se describe, por ejemplo, la secuencia de funcionamiento de los cauces. Y por último, el sexto modelo (micro-arquitectural) define las operaciones que se van a realizar en cada una de las unidades funcionales.

Como se ha dicho anteriormente, el cuarto modelo describe el repertorio de instrucciones. Este modelo es responsable de la definición de los recursos lógicos que darán soporte a las instrucciones. Es importante realizar un diseño ajustado del repertorio de instrucciones. Si un procesador posee un repertorio de instrucciones I y, por otro lado, tiene un conjunto de recursos hardware R para dar soporte al conjunto I de instrucciones, se puede declarar un conjunto R_I de recursos hardware que da soporte a todo el repertorio de instrucciones I [10]. Dada una instrucción i del repertorio de instrucciones I, diremos que el subconjunto de recursos hardware R_i es el responsable de dar soporte a esa única instrucción i $R_i \subset R$ siendo N el número de subconjuntos.

$$R_I = \bigcup_{i=1}^{N} R_i \tag{1}$$

Si aumenta el conjunto I de instrucciones soportadas por el procesador irremediablemente el hardware R_I se incrementará también. No obstante, dentro del conjunto R de recursos hardware, algunos son comunes a todas las instrucciones del repertorio I. Lo ideal sería que el hardware común fuera usado por muchas instrucciones con el fin de minimizar el coste de implementación.

Un problema específico puede usar un conjunto determinado de instrucciones A cuyos recursos R_A están contenidos en los recursos R del procesador $R_A \subset R_I$. Si el conjunto A está contenido en I se reducirá el conjunto de recursos R_A.

$$|R_A| \ll |R_I| \tag{2}$$

En definitiva, la adecuada sintonización del repertorio de instrucciones y de los recursos hardware puede optimizar el coste y el consumo.

IV. Operador de exponenciación modular

Como caso de estudio que ilustre el ciclo de desarrollo con LISA, se describirá el hardware requerido para implementar el opcode de la exponenciación modular según el algoritmo de Montgomery (mnemonico EXMODMT opA, opE, opM, opK). tal que:

$$C \equiv A^E mod M \tag{3}$$

Siendo C,A, E y M cuatro enteros representados en binario sobre k-bits donde C,A y E pertenecen a Z_M.

Para calcular la exponenciación modular utilizaremos el algoritmo expuesto a continuación:

```
ExponenciacionModular(A,E,M,k)
{
    /* Fase 1.*/
    A1 = RepresentacionMontgomery(1,k,M);
    A2 = RepresentacionMontgomery(A,k,M);
    while E != 0
    {
        /* Fase 2.*/
        A3 = AlgoritmoMontgomery(A1,A2,M);
        A2 = AlgoritmoMontgomery(A2,A2,M);
        if E(0) == 1
        {
            A1 = A3;
        }
        E = E >> 1;
    }
    return(AlgoritmoMontgomery(A1,1,M));
}
```

La representación de Montgomery de un numero binario se computa a través de operaciones básicas de desplazamiento y resta. Por otro lado, el algoritmo de Montgomery se computa en base a dos sumas y un desplazamiento. A priori es necesario utilizar dos entidades RM idénticas trabajando en paralelo para que resuelvan estas dos representaciones de Montgomery y generen el resultado A1 y A2. Posteriormente es necesario realizar dos entidades AM para operar en paralelo de nuevo y resolver la exponenciación gracias al algoritmo de Montgomery. Para controlar la ejecución del bucle principal, desplazar y analizar el exponente es necesario una quinta entidad que trabaje de manera concurrente con respecto a las otras cuatro entidades operativas.

Cada entidad se implementa arquitecturalmente mediante segmentación aritmética encauzada. A continuación se muestra un ejemplo de especificación de recursos y declaración de etapas en LISA haciendo uso del hardware propuesto:

```
RESOURCE
{
    PIPELINE ppu_cauce_multifuncion = {LD; EXE1;
        EXE2; EXE3; ST};
    PIPELINE_REGISTER IN ppu_cauce_multifuncion {
        bit[6] OpCode;
        S32 *opA, *opE, *opM, *opK;};}

OPERATION EXMODMT{
        DECLARE { .. }
        CODING { OpCode opA opE opM opK }
        SYNTAX { "EXMODMT" opA","opE","opM","opK }
        BEHAVIOR{ ExponenciacionModular(); }}

OPERATION EXE1{
    DECLARE { ENUM op= ADD,SUB;}
    SWITCH(op){
        CASE ADD: BEHAVIOR { R[..] += R[..];}
        CASE SUB: BEHAVIOR { R[..] -= R[..];}}}

OPERATION EXE2{..}

OPERATION EXE3{
DECLARE { ENUM op= SHIFT_R, SHIFT_L;}
SWITCH(op){
    CASE SHIFT_R: BEHAVIOR { R[..] = R[..] >> R
        [..];}
    CASE SHIFT_L: BEHAVIOR { R[..] = R[..] << R
        [..];}}}
```

El objetivo principal al encauzar la resolución de estas operaciones, es conseguir un alto nivel de paralelismo y alcanzar un esquema de ejecución vectorial que permita realizar cálculos sobre operandos de k-bits. Si los registros del operador tienen una anchura de n-bits, siendo $k > n$, será necesario un total de $\frac{k}{n}$ iteraciones sobre dicho operando para conseguir computarlo en su totalidad.

Otro objetivo importante en el diseño de este criptosistema es conseguir optimizar el uso de los recursos hardware favoreciendo que puedan ser utilizados por muchos códigos de operación. Así, es posible diseñar un único tipo de entidad encauzada multifunción (configurable) para que pueda resolver la operación presentada como caso de estudio y otras adicionales. Los cauces multifunción [11] pueden efectuar diferentes operaciones variando adecuadamente la interconexión de sus etapas a la vez que pueden iterar sobre los operandos para resolver esquemas de ejecución vectorial. En la Figura 1 se muestra la arquitectura del operador.

Como se ve en la Figura, se establecen dos tipos de etapas: etapas sumadoras-restadoras y etapas de desplazamiento. Cada una de ellas necesita configurarse para establecer la operación a resolver. Dado el carácter vectorial del procesamiento de números arbitrariamente grandes, cada una de las etapas posee una bandera de cero y otra de acarreo para propagar dicha información de estado al siguiente paso de cómputo.

Para diseñar una única entidad que pueda resolver la representación de Montgomery, el algoritmo de Montgomery y

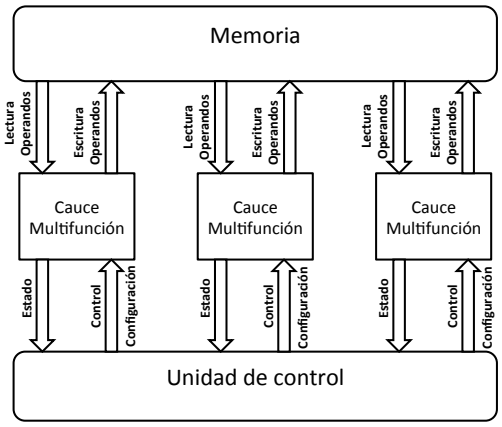


Figura 1. Arquitectura del operador

el desplazamiento del exponente, es necesario disponer de 2 bloques con operadores de suma/resta y un bloque desplazador. Cada una de estas etapas va a estar conectada entre sí mediante multiplexores M que obtienen y vuelcan los datos en 4 buses A B C y D diferentes para así variar el camino de los datos según se necesite. A su vez cada etapa tiene una línea de configuración donde se puede elegir la operación exacta a realizar (suma o resta, desplazamiento a derecha o izquierda). La etapa de load tiene como función obtener los operandos desde la memoria y la etapa store se encarga de volcar los resultados en memoria. Este cauce aritmético segmentado se puede utilizar para computar otras operaciones correspondientes a otros opcodes diseñados con LISA. En la Figura 2 se muestra la arquitectura del cauce multifunción.

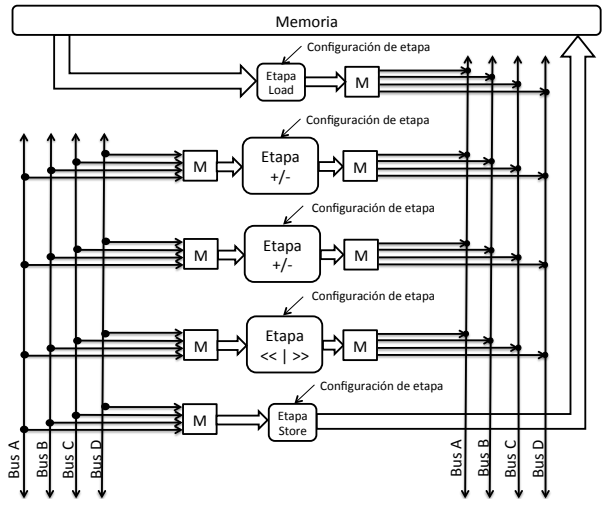


Figura 2. Arquitectura del cauce multifunción

De acuerdo al algoritmo propuesto, en la primera fase dos cauces multifunción se configuran para resolver la representación de Montgomery. Posteriormente y una vez resueltas ambas representaciones de Montgomery, se realiza el cambio de función en los cauces para que puedan resolver el algoritmo de Montgomery mientras el tercer cauce computa el desplazamiento de exponente para controlar el bucle de ejecución del algoritmo. Una vez desplazado todo el exponente se calcula una última iteración del algoritmo de Montgomery en uno de los cauces.

V. Conclusión

Los criptoprocesadores cubren necesidades de cálculo cada vez más demandadas. Hemos visto que el lenguaje LISA unifica en un solo entorno el flujo de diseño de procesadores de propósito específico (ASIP), acortando el tiempo diseño y disminuyendo los costes hardware y el consumo. La curva de aprendizaje que deben experimentar los ingenieros implicados es inicialmente abrupta pero el esfuerzo queda compensado a corto plazo por la reducción de los tiempos de desarrollo. Todo apunta a que el empleo de LISA traerá innegables beneficios al área de los criptoprocesadores.

Como caso de estudio se ha propuesto la implementación de una instrucción que calcula la exponenciación modular. El hardware asociado está basado en cauces multifunción con capacidad de procesamiento vectorial. Aunque dicho hardware da soporte al código de operación objeto del caso de estudio, también da soporte a operaciones de suma, resta y desplazamiento sobre números de tamaño arbitrario. Esta característica permite la reutilización del hardware para varios códigos de operación abaratando el coste y disminuyendo el consumo. Además, el tamaño k de los operandos es arbitrario.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Economía y Competitividad mediante el proyecto TIN2011–22668.

Referencias

[1] "Global Shipments of Cybersecurity Microcontrollers to Rise 91 Percent by 2017," IHS Online Pressromm, Report 10/7/13. Disponible en http://press.ihs.com/press-release/design-supply-chain-media/global-shipments-cybersecurity-microcontrollers-rise-91-perc.

[2] L. Bossuet, M. Grand, L. Gaspar, V. Fischer, G. Gogniat, "Architectures of Flexible Symmetric Key Crypto Engines, A Survey: From hardware Coprocessor to Multi-Crypto-Processor System on Chip," *ACM Computing Systems Surveys*, vol. 45, no. 4, Article 41, 2013.

[3] Xilinx, "CryptoBlaze: 8-Bit Security Microcontroller," XAPP374, (v1.0), 2003.

[4] S. Gueron, "Intel Advanced Encryption Standard (AES) Instructions Set," White Paper, Intel Mobility Group, Israel Development Center.

[5] G. Gogniat, T. Wolf, W. Burleson, J.P. Bossuet, R. Vaslin "Reconfigurable hardware for high-security/high-performance embedded systems: The safes perspective," en *IEEE Trans. VLSI Syst.*, 16,2, pp. 144–155.

[6] R. Buchty, N. Heintze, D. Oliva, "Criptonite - A programmable crypto processor architecture for high-bandwidth applications," en *Proc. of the Organic and Pervasive Computing Conf.*, vol 2981, Springer, 184-198.

[7] L. Gaspar, V. Fischer, F. Bernard, L. Bossuet, P. Cotret, "HCrypt: A Novel Concept of Crypto-Processor with Secured key Management," en *Int'l Conf. on Reconfigurable Computing*, Cancun, Mexico: DOI:10.1109/FPT.2011.6132722.

[8] D. Fronte, A. Perez, E. Payrat "Celator: a multi-algorithm cryptographic co-processor," en *Int'l Conf. on Reconfigurable Computing*, 2008.

[9] A. Hoffmann, H. Meyr, R. Leupers, "Architecture Exploration for Embedded Processor with LISA," *Kluwe Academic Publishers*, 2003.

[10] C. Mutigwe, J. Kinyua, F. Aghdasi, "Instruction Set Usage Analysis for Application-Specific Systems Design," *Int'l Journal of Information Technology and Computer Science*, vol. 7, no. 2, 2013.

[11] K. Hwang, F. Briggs "Arquitectura de computadoras y procesamiento paralelo," McGraw Hill, 1988.

Retos en el diseño de un generador caótico en tecnología CMOS submicrónica

Francisco Aznar
Centro Universitario de la Defensa
Grupo de Diseño Electrónico
Universidad de Zaragoza
Email: faznar@unizar.es

Carlos Sánchez-Azqueta, Cecilia Gimeno
Departamento de Ingeniería Electrónica y Comunicaciones
Grupo de Diseño Electrónico
Universidad de Zaragoza
Email: csanaz@unizar.es, cegimeno@unizar.es

***Resumen*—En este artículo se exponen los retos para llevar a cabo el diseño de un generador caótico, basado en el circuito de Chua, en tecnología CMOS submicrónica. El diseño analógico del generador caótico se complementa con un control digital, que proporciona programabilidad para definir distintos estados (claves) que aumenten la seguridad del cifrado. Además, se analizan distintas variables (temperatura, mismatching...) que pueden afectar a la sincronización de dos sistemas idealmente idénticos, impidiendo el descifrado de la información transmitida.**

***Palabras clave*—Circuito de Chua, comunicación segura, generador caótico, tecnología CMOS *(Chaotic generator, Chua's circuit, CMOS technology, secure communication)*.**

I. Nomenclatura

CMOS	Complementary Metal-Oxide Semiconductor
PSSR	Power Supply Rejection Ratio
VHF	Very High Frequency
FHSS	Frequency-Hopping Spread System

II. Introducción

Las comunicaciones representan un pilar esencial de nuestra sociedad. En particular, garantizar la privacidad se ha convertido en una temática de investigación prioritaria. Hasta el punto de que se trata de uno de los retos identificados en la Estrategia Española de Ciencia y Tecnología y de Innovación. De igual manera se consideran las TICs aplicables a este reto como una de las Tecnologías Facilitadoras Esenciales identificadas en el Programa Europeo Horizonte 2020. Esto se debe a que la seguridad de las comunicaciones es fundamental en algunas aplicaciones en el ámbito civil (transmisión de datos bancarios, datos personales...) y toma mayor relevancia en el ámbito militar. Existen multitud de sistemas de cifrado adaptados al nivel de seguridad necesario para cada aplicación concreta. Los avances en los algoritmos y potencia de cálculo de los descifradores exigen una continua mejora de los métodos de cifrado.

En el ámbito de la defensa, las comunicaciones por radio se transmiten cifradas en un rango de frecuencia reservado de la banda VHF (30 - 88 MHz). El método de cifrado es doble, por un lado se usa el FHSS, que consiste en transmitir la información sobre una onda portadora cuya frecuencia va cambiando aleatoriamente siguiendo un patrón conocido por el emisor y el receptor. Este método evita que la señal se intercepte, por lo que impide, no solo que se conozca la información transmitida, sino también la capacidad de localizar el emisor mediante triangulación. Además, la señal transmitida se envía cifrada mediante un método solo conocido por el fabricante del sistema de radio.

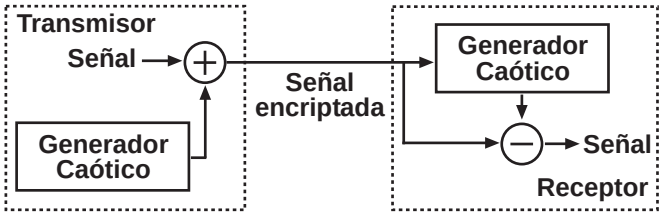


Figura 1. Esquema conceptual de una transmisión basada en cifrado caótico

Las señales caóticas, dada su naturaleza impredecible y su amplio ancho de banda, son candidatas idóneas para cifrar señales vulnerables [1]. El sistema de comunicaciones (figura 1) se basa en dos generadores caóticos idénticos y sincronizados [2]. La señal a transmitir se mezcla con la señal caótica (representado simbólicamente como una suma) y se recupera la señal transmitida mediante el desacoplo de la misma señal caótica (representado simbólicamente como una resta). Ya se ha demostrado que el cifrado caótico no es 100 % seguro [3], [4], pero incluir este sistema puede ofrecer un aumento del nivel de seguridad por complementariedad a los métodos ya implementados.

El proceso de fabricación de circuitos integrados dominante en la actualidad es el CMOS. Se basa en la construcción de los elementos activos basados en transistores MOS en una misma oblea de material semiconductor (silicio) mediante difusiones donadoras (N) y aceptoras (P), un aislante (oxido de silicio) y un material conductor (polisilicio). Además, se añaden varias capas de metalización para la interconexión de los distintos elementos (figura 2). Esta tecnología ofrece las prestaciones necesarias para trabajar en un rango de alta frecuencia con bajo consumo [5].

En este artículo se presenta el inicio de una línea de investigación que tiene por objeto desarrollar un sistema de comunicaciones seguro basado en tecnología CMOS. En la

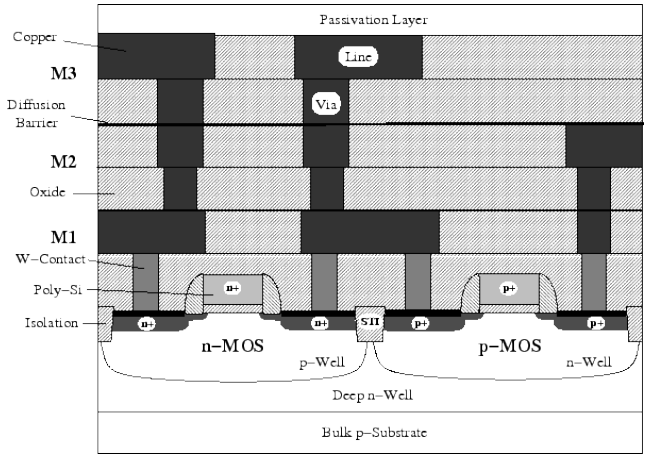


Figura 2. Sección de un proceso de fabricación CMOS con tres niveles de metalización

sección III se aborda el diseño del generador caótico en tecnología CMOS submicrónica, incorporando programabilidad para dotar al sistema de multiples claves criptográficas. En la sección IV se analizan los aspectos más relevantes que afectan al comportamiento de dos generadores caóticos teóricamente idénticos, por lo que se perdería el sincronismo, y por lo tanto no se podría recuperar la señal transmitida. Por último, las conclusiones son expuestas en la sección V.

III. Diseño del generador caótico

El generador caótico en el que nos basamos es el circuito de Chua, mostrado en la figura 3, que representa el circuito electrónico caótico más sencillo ya que cumple los requisitos mínimos necesarios: un sistema de tres ecuaciones diferenciales de primer orden, formado por L, C_1 y C_2, y un elemento no lineal (R_N). Las ecuaciones que describen la dinámica del sistema son:

$$\begin{aligned} C_1\frac{dv_1}{dt} &= \frac{1}{R}(v_1 - v_2) - f(v_1) \\ C_2\frac{dv_2}{dt} &= \frac{1}{R}(v_1 - v_2) + i_3 \\ L\frac{di_3}{dt} &= -v_2 \end{aligned} \quad (1)$$

Siendo f la función que define la resistencia no lineal. La solución de este sistema de ecuaciones diferenciales puede presentar un comportamiento caótico, tal y como se demuestra en [6].

III-A. Implementación microelectrónica del circuito de Chua

La industria microelectrónica ha realizado un increíble avance tecnológico en los últimos años. Sin embargo, dicho avance se centra básicamente en la miniaturización del transistor MOS, dotando de mejores prestaciones a los circuitos digitales fundamentalmente. Si queremos diseñar en tecnología CMOS un circuito analógico como el circuito de Chua, nos encontramos con serias limitaciones a la hora de implementar elementos tan comunes como resistencias y condensadores.

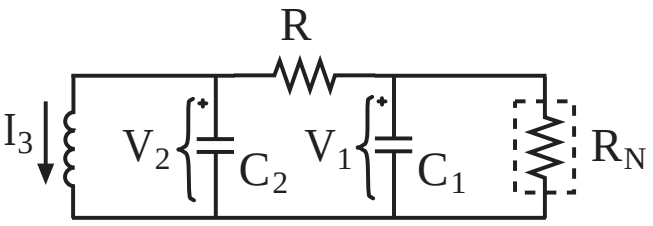


Figura 3. Circuito de Chua

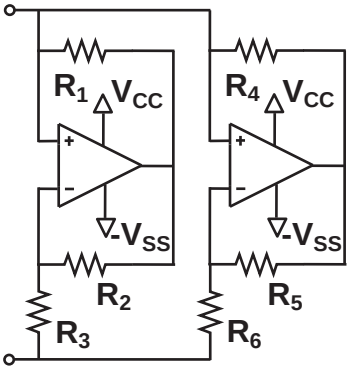


Figura 4. Implementación de la resistencia no lineal mediante amplificadores operacionales

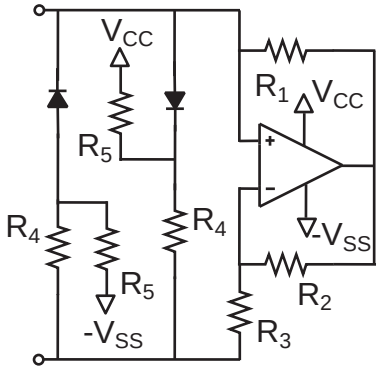


Figura 5. Implementación de la resistencia no lineal incluyendo diodos

Afortunadamente, el hecho de que para cubrir algunas aplicaciones sea imprescindible contar con electrónica analógica ha propiciado que el proceso de fabricación CMOS ofrezca la implementación de resistencias fabricadas en polisilicio, condensadores plano-paralelos entre dos de sus niveles de metalización (o bien, dos capas de polisilicio) e inductores basados en pistas metálicas con forma espiral.

El orden de magnitud de la resistencia por cuadro ($1k\Omega$) y la capacidad por unidad de área ($1fF/\mu m^2$) que ofrece un proceso CMOS de coste moderado conlleva que el área ocupada por el circuito sea considerable. Por lo tanto, para minimizar el coste final, que está directamente relacionado con

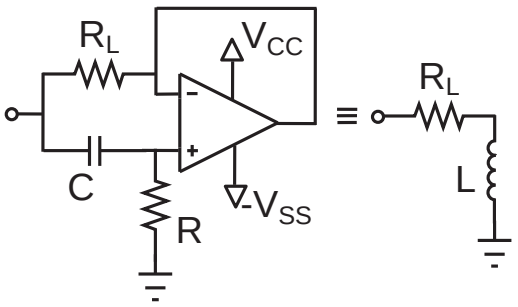


Figura 6. Implementación del inductor mediante amplificadores operacionales

el área ocupada, deberemos buscar estrategias de optimización. Pero el gran reto de la implementación microelectrónica del circuito de Chua viene de la mano de la resistencia no lineal y, especialmente, el inductor. La implementación de una resistencia no lineal se puede llevar a cabo mediante dos resistencias negativas en paralelo [7], basándonos en amplificadores operacionales (figura 4). Otra posibilidad es utilizar diodos [8] (figura 5), los cuales aportan la característica no lineal, derivado de su comportamiento asimétrico en cuanto a la conducción de la corriente. En la literatura se pueden encontrar diversos trabajos que demuestran el comportamiento caótico con elementos discretos, por ejemplo para transmisión de audio en frecuencia modulada [9]. La resistencia no lineal también se puede implementar con amplificadores de transconductancia [10], formados únicamente por transistores, lo que facilita la integración en tecnología CMOS [11], [12].

El inductor es difícil de implementar debido a su naturaleza de elemento tridimensional. Existen implementaciones en tecnología CMOS basadas en estructuras bidimensionales, pero ofrecen pobres prestaciones y contribuyen a aumentar el área del circuito significativamente (para inductancias de nH, factor de calidad limitado en torno a 10 y radios 100 μm aproximadamente). Basándonos también en amplificadores operacionales podemos emular un inductor [13] a partir de resistencias y condensadores (figura 6). El valor de la inductancia viene determinado por la expresión $L = CRR_L$. Esta alternativa optimiza el área integrada manteniendo una filosofía de integración común.

III-B. Programabilidad

Dado que los valores de los elementos que componen el sistema caótico son la clave para descifrar la transmisión, una manera de aumentar la seguridad de la comunicación es dotar al sistema de múltiples estados, ofreciendo la posibilidad de cifrar la transmisión bajo más de una clave criptográfica. O, lo que es lo mismo, implementar un circuito programable.

La programabilidad más robusta está basada en una señal digital de control formada por un número determinado de bits, los cuales ofrecen dos estados claramente definidos. Para una señal digital de n bits, dependiendo del código usado, podemos tener hasta 2^n posibles claves. Por tanto, la implementación del circuito de Chua programable tiene carácter mixto (analógico-digital).

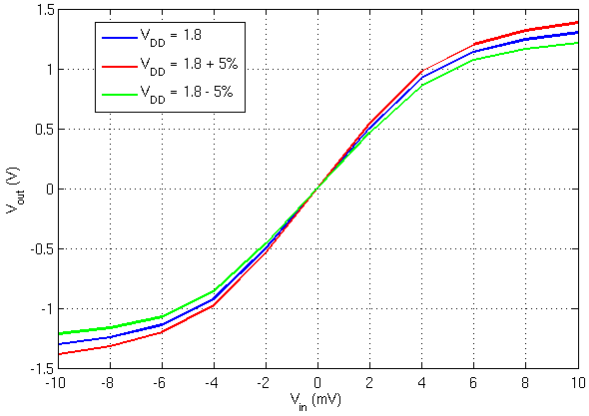


Figura 7. Variación de la relación entrada-salida para un amplificador diferencial CMOS bajo una fluctuación del $\pm 5\,\%$ en el voltaje de alimentación

La programabilidad digital está basada en el hecho de que los transistores actúan como conmutadores entre dos estados (conducción y corte) en función del estado de la señal de control asociada. Así, resultando en que uno (o varios) elementos del circuito de Chua varían su valor al formarse por varios elementos que están interconectados en paralelo cuando los transistores asociados están en conducción. En la literatura se encuentran trabajos realizados en tecnología CMOS que incluyen programabilidad digital y trabajan en rangos frecuenciales elevados [14], [15].

IV. Sincronismo

L. M. Pecora y T. L. Carroll demostraron en 1990 que dos circuitos caóticos idénticos se sincronizan [16], es decir, ofrecen un comportamiento dinámico idéntico. La señal transmitida cifrada puede actuar directamente como elemento sincronizador, como ya se reflejaba en la figura 1. El sincronismo se degrada cuando los circuitos caóticos difieren. Por lo tanto, queremos implementar un circuito programable con diferentes estados que sólo sincronicen con los estados equivalentes. Para ello debemos estudiar y analizar el impacto que tienen diferentes factores que afectan a los valores de los elementos que forman el circuito caótico. Los factores más relevantes son el voltaje de alimentación, la temperatura y la tolerancia propia de la fabricación microelectrónica.

IV-A. Voltaje de Alimentación

Los elementos activos requieren un voltaje de alimentación para su funcionamiento. La variación del valor de esa tensión puede afectar a su punto de operación, y por tanto, al comportamiento del sistema caótico. El parámetro que cuantifica la inmunidad del circuito activo frente a variaciones del voltaje de alimentación (V_{DD}) es el $PSRR$, definido como:

$$PSRR(dB) = 20log_{10}(A_V \frac{\Delta V_{DD}}{\Delta V_O}) \qquad (2)$$

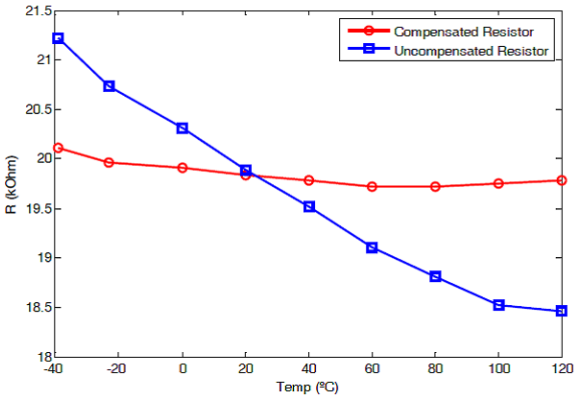


Figura 8. Variación con la temperatura para una resistencia propia de la tecnología (no compensada) y una combinación de dos resistencias con dependencia complementaria (compensada)

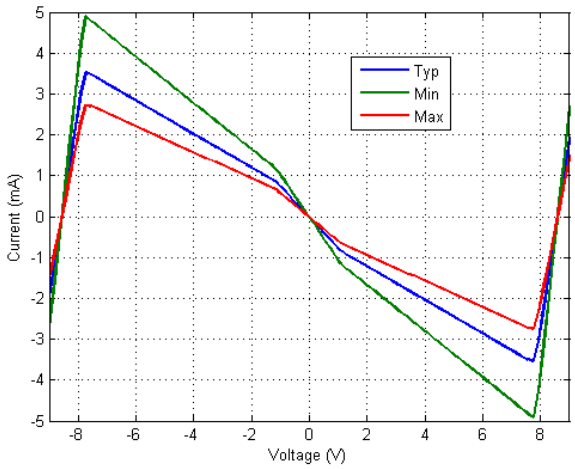


Figura 9. Curva intensidad-tensión para la resistencia no lineal para los valores nominales y corners

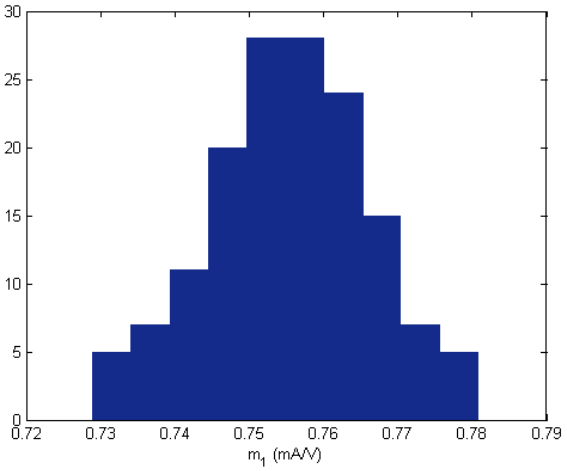


Figura 10. Distribución del valor de la pendiente para pequeña señal de la resistencia no lineal según la simulación de Montecarlo

Siendo V_O y A_V el voltaje de salida y la ganancia del circuito activo, respectivamente.

El bloque constituyente básico de un amplificador operacional es un amplificador diferencial. La figura 7 muestra la variación de la respuesta de un amplificador CMOS diferencial para una fluctuación del voltaje de alimentación de $\pm 5\,\%$. Esto lleva a un resultado en torno a $PSRR = 51dB$. Los amplificadores operacionales comerciales basados en transistores CMOS [17] ofrecen un PSRR de 75 dB como valor típico, lo que garantiza esta tecnología puede ofrecer una inmunidad sufienciente frente a variaciones del voltaje de alimentación.

IV-B. Temperatura

La temperatura de trabajo puede variar considerablemente dependiendo del entorno. Como ejemplo, el rango de temperaturas de trabajo para una circuito digital de aplicación civil cubre de 0 a 70 ºC, mientras que para aplicaciones militares se extiende de -55 a 125 ºC.

Los elementos que se ven alterados por los cambios de temperatura son las resistencias, los amplificadores de tranconductancia [18] y los diodos que forman parte de la resistencia no lineal. El valor de los condensadores se rige por factores geométricos y por el valor de la constante dieléctrica, mientras que los amplificadores operacionales se diseñan con una ganancia muy elevada, lo que ofrece inmunidad ante dicho valor por lo que el comportamiento del circuito activo viene determinado por los elementos pasivos que lo acompañan (resistencias y condensadores).

Las resistencias implementadas en tecnología microelectrónica ofrecen una dependencia con la temperatura caracterizada por el propio fabricante. Afortunadamente, existen varios tipos de resistencias que ofrecen un comportamiento con la temperatura complementario. Así, combinando ambos tipos podemos implementar resistencias con un valor muy constante en el rango de temperaturas de trabajo [19], tal y como queda reflejado en la figura 8.

Una implementación del circuito de Chua basada en amplificadores de transconductancia ofrecerá una variabilidad frente a cambios de temperatura. Los diodos ven modificada su respuesta ya que la temperatura afecta a la tensión umbral y a la corriente inversa de saturación. Este efecto podríamos considerarlo despreciable para el comportamiento del generador caótico, pero la construcción de la resistencia no lineal sin diodos ofrece un comportamiento más robusto en función de la temperatura, siendo por tanto la opción predilecta.

IV-C. Mismatching

La repetitibilidad de los valores a la hora de fabricar elementos electrónicos no esta garantizada. De hecho, las tolerancias de fabricación de elementos discretos varían entre un $10\,\%$ y un $1\,\%$. La industria microelectrónica propicia una minimización de la variabilidad de los parámetros, derivada de la fabricación simultánea de varios circuitos. Además se siguen criterios de optimización tales como respetar la orientación, no usar dimensiones mínimas y estructuras en centroide común.

Así, las tolerancias se minimizan, sobre todo para los valores relativos entre elementos fabricados en una misma oblea. Los fabricantes ofrecen modelos para el simulado de la variabilidad de los parámetros bajo criterios extremos (corner) y con dispersión gaussiana siguiendo modelos de Montecarlo.

Como ejemplo, la figura 9 muestra el efecto de las variaciones corner sobre la resistencia no lineal, fundamentalmente causadas por la variación del valor de las resistencias. Esta simulación conlleva una variación de la pendiente para pequeña señal (denominada m_1) de $-22\,\%$ y $37{,}5\,\%$ respecto del valor nominal. Estos valores están sobreestimados ya que representan casos extremos. Una simulación de Montecarlo ofrece una dispersión gaussiana (figura 10) de menos de $3\,\%$ (bajo el criterio de dos veces la desviación típica) sin implementar técnicas de minimización. Al implementar dichas técnicas, quedaría garantizada una baja dispersión de los parámetros de la resistencia no lineal.

V. Conclusiones

El factor determinante para la viabilidad de un sistema basado en cifrado caótico es el sincronismo entre transmisor y receptor. La arquitectura del generador caótico basada en resistencias, condensadores y amplificadores operacionales se postula como la más robusta. Un PSRR elevado, la compensación de la dependencia con la temperatura de la resistencia y la minimización de la dispersión causada por la fabricación del circuito integrado son los factores detectados como clave durante la etapa de diseño en un proceso CMOS submicrónico. Una tecnología CMOS actual de bajo coste ofrece altas prestanciones en términos de frecuencia y consumo. Además, se incluye la posibilidad de incluir programabilidad para aumentar la seguridad de la transmisión.

Agradecimientos

Los autores agradecen la ayuda concedida por el Centro Universitario de la Defensa (2013-12) para iniciar esta línea de investigación, asi como la colaboración del Grupo de Diseño Electrónico mediante el proyecto del Plan Nacional de I+D+i TEC2011-23211. Además, agradecer la colaboración del Capitán Javier Gil Marín, especialista en transmisiones de la Academia General Militar, por el asesoramiento ofrecido.

Referencias

[1] T. Yang, "A Survey of Chaotic Secure Communication Systems," *International Journal of Computational Cognition*, vol. 2, no. 2, pp. 81–130, 2004.

[2] A. Riaz, M. Ali, "Chaotic Communications, their Applications and Advantages over Traditional Methods of Communication," en *Proceedings of 6th International Symposium on Communications Systems, Networks and Digital Signal Processing (CNSDSP2008)*, 2008, pp. 21–24.

[3] G. Álvarez, F. Montoya, G. Pastor, M. Romera, "Breaking a secure communication scheme based on the phase synchronization of chaotic systems," *Chaos*, vol. 14, no. 2, pp. 274–278, 2004.

[4] A. B. Orúe, M.J. García-Martínez, G. Pastor, F. Montoya, C. Sánchez Ávila, "Criptoanálisis de un criptosistema de dos canales basado en una función no lineal caótica," en *Actas de la XII Reunión Española sobre Criptología y Seguridad de la Información (RECSI2012)*, U. Zurutuza, R. Uribeetxeberria, I. Arenaza-Nuño, Eds. Arrasate - Mondragon: Servicio Editorial de Mondragon Unibertsitatea, 2012, pp. 119–124.

[5] F. Aznar, S. Celma, B. Calvo, "CMOS Receiver Front-ends for Gigabit Short-Range Optical Communications," New York, NY: Analog Circuits and Signal Processing Series, Springer, 2013.

[6] L. O. Chua, G. Lin, "Canonical Realization of Chua's Circuit Family," *IEEE Transaction of Circuits and Systems*, vol. 37, no. 7, pp. 885–902, 1990.

[7] M. P. Kennedy, "Robust OpAmp Realization of Chua's Circuit," *Frequenz*, vol. 46, no. 3-4, pp. 66–80, 1992.

[8] P. Bratissol, L. O. Chua, "The double hook [nonlinear chaotic circuits]," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 12, pp. 1512–1522, 1988.

[9] P. Khumsat, G. Nowlkeaw, "Chaotic radio for audio communications," en *IEEE Region 10 Conference (TENCON2007)*, 2007, pp. 1–4.

[10] J. M. Cruz, L. O. Chua, "A CMOS IC Nonlinear Resistor for Chua's Circuit," *IEEE Transactions on Circuits and Systems - I: Fundamental Theory and Applications*, vol. 39, no. 12, pp. 985–995, 1992.

[11] A. Rodríguez-Vázquez, M. Delgado-Restituto, "CMOS Design of Chaotic Oscillators Using State Variables: A Monolithic Chua's Circuit," *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, vol. 40, no. 10, pp. 596–613, 1993.

[12] J. M. Cruz, L. O. Chua, "An IC Chip of Chua's Circuit," *IEEE Transactions on Circuits and Systems - II: Analog and Digital Signal Processing*, vol. 40, no. 10, pp. 614–625, 1993.

[13] B. Muthuswamy, T. Blain, K. Sundqvist, "A Synthetic Inductor Implementation of Chua's Circuit," Technical Report No. UCB/EECS-2009-20, Electrical Engineering and Computer Sciences, University of California at Berkeley, 2009. Disponible en http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-20.html.

[14] B. Calvo, S. Celma, F. Aznar, J.P. Alegre, "Low-voltage CMOS programmable gain amplifier for UHF applications," *Electronics Letters*, vol. 43, no. 20, pp. 1087–1088, 2007.

[15] A. Otin, S. Celma, C. Aldea, "CMOS filter with wide digitally programmable VHF range," *Electronics Letters*, vol. 43, no. 1, pp. 21–23, 2007.

[16] L. M. Pecora, T. L. Carroll, "Synchronization in Chaotic Systems," *Physical Review Letters*, vol. 64, no. 8, pp. 821–824, 1990.

[17] Texas Instruments, "250MHz, Rail-to-Rail I/O, CMOS Opertional Amplifiers," Datasheet, 2009. Disponible en http://www.ti.com/lit/ds/symlink/opa354.pdf.

[18] Texas Instruments, "LM13700 Dual Operational Transconductance Amplifiers with Linearizing Diodes and Buffers," Datasheet, 2013. Disponible en http://www.ti.com/lit/ds/symlink/lm13700.pdf.

[19] C. Azcona, B. Calvo, S. Celma, "Voltage-to-Frequency Converters," New York, NY: Analog Circuits and Signal Processing Series, Springer, 2013.

Familias de curvas elípticas adecuadas para Criptografía Basada en la Identidad

Josep M. Miret
Departament de Matemàtica
Universitat de Lleida
Email: miret@matematica.udl.cat

Daniel Sadornil
Departamento de Matemáticas, Estadística y Computación
Universidad de Cantabria
Email: sadornild@unican.es

Juan G. Tena
Departamento de Algebra, Análisis matemático y Geometría y Topología
Universidad de Valladolid
Email: tena@agt.uva.es

***Resumen*—La Criptografía Basada en la Identidad hace uso de curvas elípticas que satisfacen ciertas condiciones (*pairing-friendly curves*), en particular, el grado de inmersión de dichas curvas debe ser pequeño. En este trabajo se obtienen familias explícitas de curvas elípticas idóneas para este escenario. Dicha criptografía está basada en el cálculo de emparejamientos sobre curvas, cálculo factible gracias al algoritmo de Miller. Proponemos una versión más eficiente que la clásica de este algoritmo usando la representación de un número en forma no adyacente (*NAF*).**

***Palabras clave*—Algoritmo de Miller (*Miller's Algorithm*); Criptografía Basada en la Identidad (*Identity Based Encryption*); Curvas elípticas (*Elliptic curves*); emparejamientos (*pairings*); Forma No Adyacente (*NAF*); grado de inmersión (*embedding degree*).**

I. Introducción

Para evitar los problemas de la autentificación de las claves públicas (certificados y autoridades de certificación) que planteaba la Criptografía de Clave Pública clásica, Shamir en 1984 [12] propuso un nuevo paradigma: la Criptografía Basada en la Identidad, en la cual la clave pública de un usuario es su propio nombre o cualquier otro atributo ligado al mismo. Una de las formas en que es posible realizar la idea de Shamir es utilizando emparejamientos (pairings) sobre curvas elípticas ([5]) (otras aproximaciones basadas en el problema de la residuosidad cuadrática han sido desarrolladas por Cooks [1] o por Boneh [4]).

En lo que sigue, E representará una curva elíptica definida sobre un cuerpo finito con q elementos $\mathbb{F}_q$, $q = p^m$, p primo, $p \geq 5$, con ecuación en la forma canónica de Weierstrass $y^2 = x^3 + Ax + B$; $A, B \in \mathbb{F}_q$. Recordemos ([9]) que el conjunto $E(\mathbb{F}_q)$ de puntos de esta curva sobre $\mathbb{F}_q$ tiene cardinal $N = q + 1 - t$, con $|t| \leq 2\sqrt{q}$ y $E(\mathbb{F}_q)$ admite una estructura de grupo abeliano.

Los emparejamientos, Weil, Tate, etc, ([3]) sobre la curva E son aplicaciones bilineales con valores en un cuerpo extensión $\mathbb{F}_{q^k}$. El número k grado de inmersión, viene dado por la siguiente,

Definición 1: Sea ℓ un divisor de $N = \sharp E(\mathbb{F}_q)$ (habitualmente ℓ primo). Se denomina grado de inmersión de $E/\mathbb{F}_q$ respecto de ℓ al mínimo entero positivo k verificando las condiciones equivalentes:

i) $\ell \mid (q^k - 1)$.
ii) $\mathbb{F}_{q^k}^*$ contiene un subgrupo cíclico de orden ℓ.

Si ℓ es el mayor divisor primo de N, k se denomina simplemente grado de inmersión de $E/\mathbb{F}_q$.

Por razones computacionales, la Criptografía Basada en la Identidad requiere curvas con grado de inmersión pequeño. En particular, las denominadas curvas supersingulares (aquellas para las que $p|t$) son idóneas para este propósito ya que estas curvas tienen siempre $k \leq 6$ ([9]). Por contra, las curvas elípticas ordinarias (aquellas no supersingulares) con grado de inmersión pequeño son una minoría en el conjunto de todas las curvas posibles y su caracterización es complicada (ver [7]). Un cálculo efectivo de los emparejamientos se puede realizar usando el algoritmo de Miller ([10]).

En el presente artículo consideramos la familia de curvas elípticas con ecuación de Weierstrass $y^2 = x^3 + Ax$ cuyas propiedades han sido estudiadas ampliamente, en particular la caracterización de sus clases de isomorfía ([11]). Esta familia proporciona ejemplos de curvas tanto supersingulares como ordinarias. En la sección II se estudiará el grado de inmersión de curvas en esta familia, en el caso supersingular, su grado de inmersión puede ser obtenido fácilmente y para el caso ordinario proponemos un método para determinar las curvas con un grado de inmersión pequeño y prefijado.

En la sección III presentamos una variante del algoritmo de Miller basada en la expresión en forma no adyacente de un número natural. En la sección IV se presentan ejemplos numéricos y tiempos de ejecución de la implementación realizada del algoritmo de Miller y las variantes propuestas.

Finalmente, en la sección V se detallan las conclusiones obtenidas en la presente comunicación.

II. Curvas con grado de inmersión pequeño

En [11] se caracterizan todas las clases de isomorfía de curvas elípticas con ecuación de Weierstrass del tipo $y^2 = x^3 + Ax$.

Proposición 2: El número de clases de isomorfía de curvas elípticas de la familia mencionada sobre $\mathbb{F}_q$, $q = p^m$ viene dado por:

i) Si $q \equiv 1 \bmod 4$ entonces existen cuatro clases con representantes

$$E_i : y^2 = x^3 + \omega^i x, \quad 0 \le i \le 3,$$

donde ω es un generador de $\mathbb{F}_q^*$. Para $p \equiv 3 \bmod 4$, (por tanto m par), éstas son supersingulares. E_1, E_3 tienen cardinal $q+1$ mientras que E_0 tiene cardinal $q+1\pm 2\sqrt{q}$ y el cardinal de E_2 es $q+1\mp 2\sqrt{q}$ (el signo corresponde a $m \equiv 2, 0 \bmod 4$).

Si $p \equiv 1 \bmod 4$ las cuatro curvas son ordinarias.

i) Si $q \equiv 3 \bmod 4$ ($p \equiv 3 \bmod 4$ y m impar) entonces existen dos clases con representantes

$$E'_1 : y^2 = x^3 + x, \qquad E'_{-1} : y^2 = x^3 - x.$$

Ambas curvas son supersingulares con cardinal $q+1$.

En el caso supersingular, el grado de inmersión k de las curvas anteriores se deduce fácilmente a partir de su cardinal (ver [9]). Explícitamente, dicho grado se muestra en la tabla I.

Tabla I
GRADO DE INMERSIÓN DE CURVAS SUPERSINGULARES

Curva	k
E_0, E_2	1
E_1, E_3	2
E'_1, E'_{-1}	2

En lo que sigue consideraremos únicamente las curvas ordinarias, es decir las curvas E_i sobre $\mathbb{F}_q$, $q \equiv 1 \bmod 4$. Para caracterizar su grado de inmersión respecto de ℓ será útil el resultado siguiente ([6]).

Lema 3: Una curva elíptica E tiene grado de inmersión k con respecto de ℓ si y sólo si $t \equiv 1 + \zeta_k \bmod \ell$, con ζ_k una raíz de orden k de la unidad módulo ℓ.

Por ejemplo, si $k = 1$ se tiene que $t \equiv 2 \bmod \ell$ y si $k = 2$ se tiene que $t \equiv 0 \bmod \ell$. Por otra parte, ℓ debe dividir tanto a $q+1-t$ como a $q^k - 1$. Fijado el grado de inmersión, para cada primo ℓ, se pueden obtener condiciones para entero q (primo o potencia de un primo).

En particular para $k = 1$ y 2 hemos demostrado que:

Teorema 4: Una de las cuatro curvas E_i tiene grado de inmersión 1 con respecto de ℓ si y sólo si

$$q = (x^2 + y^2)\ell^2 + 2x\ell + 1, \quad x, y \in \mathbb{Z},\ x \equiv y \bmod 2.$$

Teorema 5: Una de las cuatro curvas E_i tiene grado de inmersión 2 con respecto de ℓ si y sólo si

$$q = x^2\ell^2 + y\ell - 1,\ x \equiv y \bmod 2 \text{ y } y\ell - 1 \text{ es un cuadrado.}$$

Nótese que tal q debe ser primo o potencia de primo. La curva concreta puede obtenerse teniendo en cuenta la Proposición 3.5 de [11].

Ejemplos concretos para algunos valores de los parámetros x, y pueden verse en la tabla II.

Tabla II
CURVAS ELÍPTICAS CON GRADO DE INMERSIÓN 1 Ó 2

ℓ	x, y	q	Curva	k
73	0,2	$4\ell^2 + 1 = 21317$	E_0	1
41	2,2	$8\ell^2 + 4\ell + 1 = 13613$	E_2	1
79	1,1	$2\ell^2 + 2\ell + 1 = 12641$	$E_1\ (E_3)$	1
101	1,1	$\ell^2 + \ell + 1 = 10301$	E_0	2
101	3,1	$9\ell^2 + \ell - 1 = 91909$	E_2	2
1013	2,2	$4\ell^2 + 2\ell - 1 = 4106701$	$E_1\ (E_3)$	2

III. ALGORITMO DE MILLER CON FORMAS NO ADYACENTES

Como se ha dicho en la introducción, un emparejamiento de orden ℓ, e_ℓ, para una curva elíptica E sobre el cuerpo finito $\mathbb{F}_q$ (usualmente ℓ primo y divisor del cardinal de la curva) es una aplicación bilineal que asigna a un par de puntos P, Q de la curva una raíz de orden ℓ de la unidad. Supondremos que estas raíces se encuentran inmersas en $\mathbb{F}_{q^k}$ siendo k el grado de inmersión de la curva (ver definición 1). En el caso particular del emparejamiento de Weil, los puntos P, Q son ambos de ℓ-torsión (designaremos por $E(\mathbb{F}_{q^k})[\ell]$, el subgrupo de puntos de ℓ-torsión de E). Nótese que un punto de ℓ-torsión no necesariamente está definido en el cuerpo base pero si en $\mathbb{F}_{q^k}$ ([2]).

Para los puntos P, Q, el valor del emparejamiento e_ℓ se calcula como el cociente

$$e_\ell(P, Q) = \frac{f_{\ell,P}(Q + R) f_{\ell,Q}(S)}{f_{\ell,P}(R) f_{\ell,Q}(P + S)}, \tag{1}$$

donde R, S son puntos auxiliares de la curva y las funciones $f_{\ell,P}, f_{\ell,Q}$ son cociente de dos polinomios en dos variables (para más detalles ver [9]). La dificultad de este cálculo reside en la construcción de estas funciones. El algoritmo de Miller ([10]) permite realizar dicha construcción de forma eficiente. Las funciones $f_{\ell,T}$ para un punto cualquiera T, se obtienen de forma recursiva a partir de las siguientes identidades:

$$\begin{aligned} f_{0,T} &= f_{1,T} = 1 \\ f_{m+n,T} &= f_{m,T} f_{n,T} g_{mT,nT} \end{aligned} \tag{2}$$

donde $g_{U,V} = \frac{L_{U,V}}{L_{(U+V),-(U+V)}}$ y $L_{U,V} = 0$ es la ecuación de la recta que pasa por los dos puntos U y V, o la recta tangente si $U = V$. Nótese que $L_{(U+V,-(U+V)}$ es la recta vertical que pasa por el punto $U+V$ y por tanto sólo depende de su abscisa.

Por tanto, dado un entero ℓ y un punto T de orden ℓ, el cálculo de la función $f_{\ell,T}$ se realiza de la siguiente forma:

Algoritmo 6:
Input: $T \in E(\mathbb{F}_{q^k})[\ell]$,
$\ell = (\ell_{r-1}, \ell_{r-2}, \ldots, \ell_0)_2$ (representación binaria de ℓ)
Output: $f_{\ell,T}$
$f \leftarrow 1$, $W \leftarrow P$.
for i from $r-2$ to 0 **do**
$f \leftarrow f^2 \frac{L}{L'}$
(L recta tangente en W, L' recta vertical por $2W$)

$W \leftarrow 2W$
if $\ell_i = 1$ **then**
$f \leftarrow f\frac{L}{L'}$
(L recta que pasa por T y W, L' recta vertical por $T+W$)
$W \leftarrow T+W$
end if
end for
RETURN f

El algoritmo anterior se basa en una estrategia de cuadrados repetidos (doblado y suma) en la que se hace uso de la expresión binaria del entero ℓ. Es claro que el número de iteraciones en el algoritmo anterior es $log_2(\ell)$ y el número de *operaciones* (cálculo de rectas) a realizar depende del peso de Hamming de ℓ. Por tanto, el algoritmo será más rápido cuanto menor sea tal peso.

Los algoritmos que se basan en este tipo de estrategia pueden adaptarse para usar, en lugar de la representación binaria, cualquier otra cadena de adición-sustracción. En particular, es posible minimizar el número de bits no nulos en la representación binaria de un número usando el siguiente concepto en el que se permiten restas (hay que tener en cuenta que la resta de puntos de una curva elíptica se puede hacer con el mismo coste que una suma).

Definición 7 ([8]): Dado un entero n, se define la forma no adyacente de n ($NAF(n)$) como una representación dada por $n = \sum_{i=0}^{r-1} k_i 2^i$ donde $k_i \in \{0, \pm 1\}$, $k_{r-1} \neq 0$ y para todo $1 \leq i \leq r-1$, $k_{i-1}k_i = 0$.

A partir de las formas recurrentes dadas en (2), teniendo en cuenta que $f_{0,T} = f_{1,T} f_{-1,T} g_{T,-T}$, es posible determinar los pasos a añadir en el Algoritmo 6 cuando se cambia la representación binaria del entero ℓ por su representación en forma no adyacente. Más concretamente, habría que añadir las órdenes siguientes:

if $\ell_i = -1$ **then**
$f \leftarrow f\frac{L}{L_1 L'}$
(L_1 recta vertical por T, L recta que pasa por $-T$ y W, L' recta vertical por $W-T$)
$W \leftarrow W-T$
end if

Sin embargo, existe otra forma de obtener la función $f_{\ell,T}$ a partir de la representación NAF de ℓ. Para ello, se sustituirían las instrucciones anteriores por las siguientes:

if $\ell_i = -1$ **then**
$f \leftarrow f\frac{L'}{L}$
(L' recta vertical por W y L la recta que pasa por los puntos T y $-W$)
$W \leftarrow W-T$
end if

Téngase en cuenta que en este caso cuando $\ell_i = -1$, se calculan dos rectas mientras que en la versión anterior son necesarias tres rectas. Aunque la forma natural de trasladar el algoritmo clásico de Miller a su versión usando la representación NAF es la propuesta primera, la búsqueda de una solución computacionalmente más eficiente nos ha llevado a esta segunda forma.

Tenemos entonces, tres formas diferentes para el cálculo de la función $f_{\ell,T}$: método binario usando el algoritmo original de Miller y dos versiones NAF I y NAF II utilizando la representación en forma no adyacente. En realidad, la función obtenida en cada caso es diferente, sin embargo, al calcular el emparejamiento con la fórmula (1), el resultado obtenido con cualquiera de las tres funciones es el mismo, ya que dicho valor $e_\ell(P,Q)$ es independiente del camino seguido ([3]).

IV. Ejemplos numéricos

Se han implementado las diferentes versiones mostradas del algoritmo de Miller para el cálculo de la función $f_{\ell,T}$ usando Maple v.13 en un ordenador con un procesador Intel Core 2 Duo de 2.13 GHz y 2 GB de memoria RAM.

En la tabla III se muestran los tiempos de ejecución (en media) de los tres algoritmos: es decir, usando la representación binaria de ℓ en el primer caso, y en los otros dos tomando la forma no adyacente de ℓ.

Se han calculado previamente curvas elípticas con grado de inmersión 1 respecto a algún primo ℓ de tamaño 20, 40, 60 y 100 bits (con sus correspondientes puntos de orden ℓ) para las dos primeras familias de curvas elípticas ordinarias mostradas en la Tabla II.

El primo ℓ se ha escogido de dos formas distintas. En el primer caso no se ha considerado ninguna restricción sobre él mientras que en el segundo caso, se han escogido primos ℓ cuya representación en forma no adyacente tiene peso de Hamming pequeño. Para cada tipo se han tomado 500 de estos primos para cada longitud (100 cuando ℓ tiene 100 bits).

Tabla III
Tiempos de ejecución del algoritmo de Miller (ms)

Curva/primo	Nº de bits ℓ	Binario	NAF I	NAF II
E_0, $p = 4\ell^2 + 1$	20	4.1	3.8	3.8
	40	12.1	10.6	10.6
	60	23.5	21.9	20.7
	100	61.4	56.3	54.1
E_0, $p = 4\ell^2 + 1$ $w_{NAF}(\ell) \leq 7$	20	3.7	3.4	3.2
	40	10.4	8.7	8.7
	60	21.8	17.6	17
	100	56.3	44.9	43.6
E_2, $p = 8\ell^2 + 4\ell + 1$	20	6.3	6	5.9
	40	11.2	10.1	9.8
	60	33.8	30.7	28.2
	100	65.6	58.8	56.4
E_2, $p = 8\ell^2 + 4\ell + 1$ $w_{NAF}(\ell) \leq 8$	20	5.5	5.1	4
	40	13.8	12.3	11.2
	60	29.4	23.8	22.5
	100	80.9	60.2	59.9

Tal como era de esperar, en ambos casos las versiones basadas en formas no adyacentes son más eficientes y, en general, la versión NAF II es mucho más rápida.

El mayor descenso en tiempo de ejecución se encontraría cuando ℓ es un primo de Mersenne, 2^p-1, cuya representación binaria es $111\ldots111$, mientras que su expresión en forma no adyacente es $100\ldots000$-1.

Finalmente, hemos considerado la curva elíptica E_0 sobre $\mathbb{F}_q$, con $q = 4\ell^2 + 1$ siendo $\ell = 2^{258} + 2^{242} - 1$. Este primo ℓ es un ejemplo de los denominados primos de Solinas, recomendados para su uso en criptografía elíptica. Las expresiones binaria y NAF de este primo ℓ son:

$$1000000000000000\overbrace{111\ldots11}^{242bits}$$
$$1000000000000000\underbrace{1000\ldots0\text{-}1}_{243bits}$$

Los tiempos de ejecución de las tres versiones del algoritmo de Miller para este ejemplo son 468 ms para el caso binario, 234 ms para NAF I y 218 ms para NAF II.

V. Conclusión

Partiendo de las curvas con el denominado j-invariante igual a 1728, o lo que es lo mismo las curvas con ecuación $y^2 = x^3 + Ax$, hemos obtenido condiciones que permiten asegurar que dichas curvas son buenas para su empleo en Criptografía Basada en la Identidad. En particular, hemos proporcionado ejemplos concretos de curvas elípticas con grado de inmersión 1 y 2.

La herramienta principal utilizada en Criptografía Basada en la Identidad son los emparejamientos. Nos hemos ocupado también de la implementación del algoritmo de Miller que es el algoritmo básico en el cálculo de los emparejamientos, tanto en su versión original como en variantes propuestas en el trabajo basadas en la forma no adyacente del primo ℓ. De esta forma, se observa que las variantes propuestas, permiten un ahorro significativo en los tiempos de ejecución del algoritmo de Miller.

Agradecimientos

Este trabajo ha sido financiado por el Ministerio de Economía y Competitividad con los proyectos MTM2010-21580-C02-01/02 y MTM2010-16051.

Referencias

[1] C. Cocks, "An identity based encryption scheme based on quadratic residues", *Proceedings of the 8th IMA Int. Conf. on Cryptology and Coding*, Springer LNCS 2260, pp. 360–363, 2001.

[2] R. Balasubramanian, N. Koblitz "The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm",*Journal of Cryptology* vol 11 (2), pp. 141–145, 1998.

[3] I. Blake, G. Seroussi, N. Smart, "Advances in Elliptic Curve Cryptography". London Mathematical Society, LNS 371. Cambdridge University Press, 2005.

[4] D. Boneh, C. Gentry, M. Hamburg. "Space-Efficient Identity Based Encryption without Pairings" *FOCS'07: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pp. 647-657, 2007.

[5] D. Boneh, M. Franklin "Identity based Encryption from the Weil Pairing" *Advances in Cryptology - CRYPTO 2001* , Springer LNCS 2139, pp. 213–229, 2001.

[6] F. Brezing, A. Weng, "Elliptic curves suitable for pairings based cryptography", *Designs, Codes and Cryptography* 37, pp. 133–141, 2005.

[7] D. Freeman, M. Scott, E. Teske, "A taxonomy of pairing-friendly elliptic curves", *Journal of Cryptology*, vol 23(2), pp. 224–280, 2010.

[8] D. Hankerson, A. Menezes, S. Vanstone, "Guide to Elliptic Curve Cryptography", Springer, 2004.

[9] A. Menezes, "Elliptic Curves Public Key Cryptography" , Kluwer, 1993.

[10] V. Miller "The Weil Pairing, and its Efficient Calculation" *Journal of Cryptology* , vol. 17, pp. 235–261, 2004.

[11] C. Munuera, J. Tena, "An algorithm to compute the number of points on elliptic curves of j-invariant 0 or 1728 over a finite field", *Rendiconti del Circolo Matematico di Palermo*, Serie 2, tomo XLII, pp. 106–116, 1993.

[12] A. Shamir, "Identity-based cryptosystems and signature schemes", *Advances in cryptology - CRYPTO 1984*, Springer LNCS196, pp. 47–53, 1985.

Códigos con propiedades de localización basados en matrices de bajo sesgo

José Moreira
Departament d'Enginyeria Telemàtica
Universitat Politècnica de Catalunya
Email: jose.moreira@entel.upc.edu

Marcel Fernández
Departament d'Enginyeria Telemàtica
Universitat Politècnica de Catalunya
Email: marcel@entel.upc.edu

Grigory Kabatiansky
Institute for
Information Transmission Problems
Russian Academy of Sciences
Email: kaba@iitp.ru

***Resumen*—En este artículo presentamos una construcción explícita de un código con propiedades de identificación de traidores, aplicable a entornos de fingerprinting. Nuestro trabajo parte del estudio de una familia de códigos conocidos como *códigos separables*, que en el campo del fingerprinting también se conocen como *códigos seguros contra incriminaciones*. A partir de estos códigos, nos centramos en una versión menos estricta de ellos, en los que no se requiere que la propiedad de separación se cumpla en todos los casos, sino con alta probabilidad. Este tipo de códigos se conocen como *códigos cuasi seguros contra incriminaciones*. En este trabajo mostramos como construir explícitamente estos códigos, basando nuestras construcciones en estructuras conocidas como *matrices de bajo sesgo*. Además, mostramos cómo es posible utilizar dichos códigos para construir de forma explícita una familia de códigos binarios con propiedades de identificación, baja tasa de error y decodificación eficiente.**

***Palabras clave*—Fingerprinting, código seguro contra incriminaciones (*secure frameproof code*), código separable (*separating code*), identificación de traidores (*traitor tracing*)**

I. Introducción

Los códigos con propiedades de localización, también conocidos como códigos de fingerprinting, se utilizan para luchar contra la redistribución ilegítima de contenidos, llevada a cabo por usuarios deshonestos. Un distribuidor que desee proteger un determinado contenido entregará copias marcadas de éste a los usuarios destinatarios. Cada marca introducida identificará a un único usuario. Esto los disuadirá de realizar una redistribución "ingenua" de su copia del contenido. No obstante, puede suceder que diversos usuarios (traidores) confabulen y generen una copia pirata, que no es más que una mezcla, de acuerdo a unas determinadas reglas, de sus propias copias. La copia pirata, por tanto, contendrá una marca corrupta. Por tanto, el objetivo del distribuidor consistirá en determinar un conjunto de marcas tales que sea posible identificar, al menos, a uno de los traidores.

El término de *código seguro contra incriminaciones (secure frameproof code, código SFP)* [1], [2], [3], [4] es el nombre que se dio a los *códigos separables* [5], [6], [7], [8], [9], [10], [11] cuando fueron redescubiertos dentro de los campos del fingerprinting y de la identificación de traidores. Este artículo versa sobre la construcción de lo que denominamos *códigos cuasi seguros contra incriminaciones (almost secure frameproof code, código cuasi SFP)* y su aplicación a la construcción explícita de códigos de fingerprinting. Estos códigos, una versión menos restrictiva de los códigos SFP, fueron introducidos en [12]. En ese trabajo, se mostró su aplicación a la construcción de códigos de fingerprinting, al estilo de [13], mejorando las cotas de existencia previas de dichos códigos. A efectos prácticos, la idea principal consiste en que el reemplazo de códigos SFP por códigos cuasi SFP en ese tipo de construcciones permite al distribuidor utilizar códigos de fingerprinting más cortos, reduciendo así tanto el coste de inserción de las marcas como el coste de identificación de traidores [12], [14].

Sea C un código. Informalmente, diremos que dos conjuntos disjuntos de palabras código $U, V \subseteq C$ son *separados* si existe una posición en la que el conjunto de valores de las palabras de U es disjunto al conjunto de valores de las palabras de V en esa posición. El código C se denomina (c, c)-separable [5], [6], [7], [8], [9], [10], [11] si cada par de conjuntos disjuntos $U, V \subseteq C$ de tamaño c son separados.

Supongamos que, dado un conjunto $U \subseteq C$ de $\leq c$ palabras código, generamos una nueva palabra en la que el valor en cada posición pertenece a alguna de las palabras de U en esa posición. Una palabra generada de esta forma se denomina *descendiente* del conjunto U. Dado que las palabras código corresponden a las marcas de usuario, la generación de un descendiente modela la generación de la marca corrupta de la copia pirata. Un descendiente de U es *unívocamente c-decodificable* si no es descendiente de cualquier otro conjunto disjunto a U de $\leq c$ palabras código. Un código c-SFP es aquel en el que todos los descendientes de conjuntos de $\leq c$ palabras son unívocamente c-decodificables [1], [2], [3], [4]. No es difícil ver que esto es equivalente a la condición descrita para los códigos (c, c)-separables. Es decir, un código (c, c)-separable y un código c-SFP son el mismo concepto.

Consideremos ahora una versión menos estricta de ambas definiciones, en el sentido de no exigir separabilidad completa ni decodificación unívoca completa. Esto nos lleva a considerar dos nociones diferentes, como se expuso en [12]. Un código *cuasi (c, c)-separable* es un código en el que un subconjunto de $\leq c$ palabras código está separado del resto de subconjuntos disjuntos de tamaño $\leq c$ con alta probabilidad. Por otra parte, un código *cuasi c-SFP* es un código en el que cada descendiente es unívocamente c-decodificable con alta probabilidad.

En este artículo conectaremos los conceptos definidos anteriormente con el concepto de *matrices de bajo sesgo* [15], que está estrechamente relacionado con el concepto de espacios

probabilísticos de bajo sesgo [16], [17].

Una *matriz binaria de bajo sesgo* es una matriz definida sobre el cuerpo finito de dos elementos, $\mathbb{F}_2 = \{0,1\}$, tal que cualquier combinación lineal de sus columnas tiene, aproximadamente, el mismo número de ceros que de unos.

Bajo unas determinadas condiciones, una matriz binaria de bajo sesgo $A \in \mathbb{F}_2^{n\times M}$ exhibirá la siguiente propiedad: para cualquier subconjunto S de $\leq t$ columnas y cada posible vector $\mathbf{a} \in \mathbb{F}_2^t$, existirá una fila tal que su proyección en las columnas de S coincidirá con $\mathbf{a}$. Una matriz con esta propiedad genera inmediatamente lo que se conoce como un *conjunto (M,t)-universal*. Esta observación será clave para nuestros propósitos, ya que un conjunto $(M,2c)$-universal genera inmediatamente un código (c,c)-separable, es decir, c-SFP.

Como se ha comentado, es fácil ver que un código (c,c)-separable y un código c-SFP son el mismo concepto. No obstante, cuando se consideran sus versiones relajadas ambas nociones difieren. Intuitivamente, parece claro que un código cuasi separable es más restrictivo que un código cuasi SFP. Concretamente, se ha mostrado la existencia de códigos cuasi SFP de tasa mucho mayor que códigos cuasi separables [12]. La estrategia utilizada para establecer dichas cotas de existencia está basada en métodos probabilísticos que, desafortunadamente, no son métodos constructivos que nos lleven a la construcción práctica de estos códigos.

Introducidos estos conceptos, podemos dar una visión general de la estructura de este artículo. En la Sección II mostramos las definiciones formales que necesitaremos, así como una breve revisión de resultados anteriores. Nuestra contribución la presentaremos en la Sección III. Mostraremos como, partiendo de construcciones existentes de matrices de bajo sesgo, podemos obtener construcciones de lo que denominaremos conjuntos cuasi universales. Finalmente, mostramos como esas construcciones pueden emplearse para la construcción explícita de códigos cuasi SFP, lo que culminará con la construcción explícita de un código de fingerprinting en la Sección IV.

II. Definiciones y resultados previos

Dado un alfabeto Q de tamaño $|Q| = q$, denotamos por Q^n el conjunto de todos los vectores q-arios de longitud n. Por ejemplo, $\mathbf{u} = (u_1,\ldots,u_n) \in Q^n$. Un subconjunto $C \subseteq Q^n$ de tamaño M se denomina un *(n,M)-código q-ario*. Los elementos de C se denominan *palabras código*. Si Q es un cuerpo finito de q elementos, lo denotaremos por $\mathbb{F}_q$.

II-A. Códigos cuasi separables y cuasi SFP

Dado un (n,M)-código C, un subconjunto $U = \{\mathbf{u}^1,\ldots,\mathbf{u}^c\} \subseteq C$ de tamaño c se denomina *c-coalición*. Denotaremos por $P_i(U)$ la *proyección* de U en la posición i-ésima, es decir, el conjunto de elementos del alfabeto del código en dicha posición,

$$P_i(U) \stackrel{\text{def}}{=} \{u_i^1,\ldots,u_i^c\}.$$

Dadas dos c-coaliciones $U,V \subseteq C$, diremos que U y V están *separadas* si $P_i(U)\cap P_i(V) = \emptyset$ para alguna posición i. Llamaremos a esa posición una *posición separadora*. También diremos que una c-coalición U es *separada* si está separada de cualquier otra c-coalición del código.

Definición 1: Un código C es *(c,c)-separable* si cualquier par de c-coaliciones $U,V \subseteq C$ tienen una posición separadora. Equivalentemente, si todas las c-coaliciones $U \subseteq C$ son separadas.

Los códigos separables fueron introducidos por Friedman *et al.* en [5] hace más de 40 años. Un código separable es una estructura combinatoria con multitud de aplicaciones, como por ejemplo en la construcción de funciones de hash, testeo de circuitos combinacionales y síntesis de autómatas. Estos códigos han sido posteriormente estudiados por numerosos autores, por ejemplo en [6], [7], [8], [9], [10], [11]. Se han investigado cotas superiores e inferiores sobre su tasa, y se ha mostrado su relación con conceptos matemáticos similares. Véase, por ejemplo, [6] y [10].

Con la aparición del fingerprinting digital, los códigos separables han vuelto a suscitar interés de nuevo. Consideremos una c-coalicón $U = \{\mathbf{u}^1,\ldots,\mathbf{u}^c\} \subseteq C$ de un (n,M)-código $C \subseteq Q^n$. En un ataque de confabulación, las *reglas de marcado (marking assumption)* [18] establecen que las posiciones i tales que todas las palabras de U tienen el mismo símbolo deben permanecer inalteradas en cualquier palabra pirata $\mathbf{z}$ que generen. En concreto, el modelo de generación de palabras pirata que adoptaremos será el conocido como *narrow-sense envelope model* [13], donde para cada posición i tenemos que $z_i \in P_i(U)$. El conjunto de todas las palabras piratas que U puede generar lo denotaremos por $\mathrm{desc}(U)$, y bajo las presuposiciones mencionadas, tenemos que

$$\mathrm{desc}(U) \stackrel{\text{def}}{=} \{\mathbf{z} = (z_1,\ldots,z_n) \in Q^n : z_i \in P_i(U),\ 1 \leq i \leq n\}.$$

Normalmente, a las palabras pirata de $\mathrm{desc}(U)$ se les denomina *descendientes*. También se define el *código c-descendiente* de C, denotado por $\mathrm{desc}_c(C)$, como

$$\mathrm{desc}_c(C) \stackrel{\text{def}}{=} \bigcup_{U\subseteq C, |U|\leq c} \mathrm{desc}(U).$$

Un descendiente $\mathbf{z} \in \mathrm{desc}_c(C)$ es *unívocamente c-decodificable* si $\mathbf{z} \in \mathrm{desc}(U)$ para alguna c-coalición $U \subseteq C$, y $\mathbf{z} \notin \mathrm{desc}(V)$ para cualquier c-coalición V disjunta a U.

Definición 2: Un código C es *c-seguro contra incriminaciones* (c-SFP) si para cualesquiera $U,V \subseteq C$ tales que $|U| \leq c$, $|V| \leq c$ y $U \cap V = \emptyset$, entonces $\mathrm{desc}(U) \cap \mathrm{desc}(V) = \emptyset$. Equivalentemente, si todos los descendientes $\mathbf{z} \in \mathrm{desc}_c(C)$ son unívocamente c-decodificables.

El concepto de código c-SFP fue introducido en [18], [1], [2]. No es difícil ver que, en efecto, se trata de códigos (c,c)-separables.

Sea $R = R(C) \stackrel{\text{def}}{=} n^{-1}\log_q|C|$ la *tasa* de un (n,M)-código sobre un alfabeto q-ario Q. Denotaremos por $R_q(n,c)$ a la tasa máxima que puede alcanzar un código q-ario (c,c)-separable (equivalentemente, c-SFP) de longitud n. Es decir,

$$R_q(n,c) \stackrel{\text{def}}{=} \max_{\substack{C\subseteq Q^n:\, C \text{ es}\\ (c,c)\text{-separable}}} R(C).$$

Consideraremos también los límites asintóticos de dicha tasa

$$\underline{R}_q(c) = \liminf_{n\to\infty} R_q(n,c), \quad \overline{R}_q(c) = \limsup_{n\to\infty} R_q(n,c).$$

En este artículo nos centraremos en códigos sobre el alfabeto binario, es decir, $Q = \{0,1\}$. Para códigos binarios $(2,2)$-separables, se sabe que $\underline{R}_2(2) \geq 0,0642$ [7], [6] (cota también válida para el caso de códigos lineales [7]) y $\overline{R}_2(2) < 0{,}2835$ [6], [9]. Para valores arbitrarios de c, en [13] se obtiene que

$$\underline{R}_2(c) \geq -\frac{\log_2(1-2^{-2c+1})}{2c-1}.$$

Como puede observarse, las cotas de existencias de códigos separables muestran que éstos poseen una tasa muy baja. Con el objetivo de obtener códigos de mejor tasa, en [12] se proponen dos versiones menos estrictas de estos códigos.

Definición 3: Un código $C \subseteq Q^n$ es *ε-cuasi (c,c)-separable* si la proporción de coaliciones separadas de tamaño c, entre todas las posibles coaliciones de tamaño c, es $\geq 1-\varepsilon$.

Una secuencia de códigos $(C_i)_{i\geq 1}$ de longitud n_i creciente es una *familia asintóticamente cuasi (c,c)-separable* si cada código C_i es un código ε_i-cuasi (c,c)-separable y $\lim_{i\to\infty} \varepsilon_i = 0$.

Definición 4: Un código $C \subseteq Q^n$ es *ε-cuasi c-SFP* si la proporción de descendientes $\mathbf{z} \in \mathrm{desc}_c(C)$ unívocamente c-decodificables es $\geq 1-\varepsilon$.

Una secuencia de códigos $(C_i)_{i\geq 1}$ de longitud n_i creciente es una *familia asintóticamente cuasi c-SFP* si cada código C_i es un código ε_i-cuasi c-SFP y $\lim_{i\to\infty} \varepsilon_i = 0$.

Cabe destacar que las definiciones anteriores permiten separar los conceptos de "separación" y "decodificación unívoca", que coincidían en el caso de códigos completamente separables y SFP. Además, estas nuevas definiciones permiten obtener códigos con mayor tasa.

Para una familia de códigos $\mathcal{C} = (C_i)_{i\geq 1}$ definimos su tasa asintótica como

$$R(\mathcal{C}) \stackrel{\text{def}}{=} \liminf_{i\to\infty} R(C_i).$$

Nuestro interés reside en estimar el valor máximo de dicha tasa entre todas las familias de códigos asintóticamente cuasi (c,c)-separables y asintóticamente c-SFP. Denotaremos estas tasas asintóticas por $R_q^{\text{sep}*}(c)$ y $R_q^{\text{SFP}*}(c)$, respectivamente.

Por ejemplo, para el caso binario y coaliciones de tamaño $c=2$ tenemos que $R_2^{\text{sep}*}(2) \geq 0{,}1142$, de [14], y $R_2^{\text{SFP}*}(2) \geq 0{,}2075$, de [12].

II-B. Matrices de bajo sesgo

En esta sección presentamos los conceptos sobre matrices de bajo sesgo que utilizaremos más adelante en nuestras construcciones. Para una explicación más detallada, remitimos al lector a las referencias [17], [16], [15].

Como se ha comentado, nos centraremos en el caso binario, ya que nuestro objetivo final será la construcción de códigos binarios. Por tanto, de aquí en adelante trabajaremos con el alfabeto $\mathbb{F}_2$.

Una *(n,M)-matriz binaria* A es una matriz de tamaño $n \times M$ donde sus elementos pertenecen a $\mathbb{F}_2$. Dada una (n,M)-matriz binaria A y un subconjunto de posiciones $S \subseteq \{1,\ldots,M\}$ de tamaño s, Denotamos por $\nu_S(\mathbf{a};A)$ al número de filas de A cuyas proyecciones en las posiciones de S coincide con el vector $\mathbf{a} \in \mathbb{F}_2^s$. Obviamente, un vector $\mathbf{u} \in \mathbb{F}_2^n$ puede verse como una $(n,1)$-matriz binaria. En este caso, $\nu_{\{1\}}(0;\mathbf{u})$ y $\nu_{\{1\}}(1;\mathbf{u})$ denotan el número de ceros y el número de unos de $\mathbf{u}$, respectivamente.

Definición 5: Sea $\mathbf{u} = (u_1,\ldots,u_n) \in \mathbb{F}_2^n$. El *sesgo* del vector $\mathbf{u}$ se define como

$$n^{-1}|\nu_{\{1\}}(0;\mathbf{u}) - \nu_{\{1\}}(1;\mathbf{u})|.$$

Es decir, un vector $\mathbf{u}$ con, aproximadamente, el mismo número de ceros y de unos tendrá bajo sesgo.

Definición 6: Sea $0 \leq \varepsilon < 1$. Una (n,M)-matriz binaria A es *(ε,t)-sesgada* si cualquier combinación lineal no trivial de $\leq t$ de sus columnas tiene sesgo $\leq \varepsilon$. Si $t = M$ diremos simplemente que A es ε-sesgada (o que tiene sesgo ε).

Definición 7: Sea $0 \leq \varepsilon < 1$. Una (n,M)-matriz binaria A es *ε-cuasi t-independiente* si para cualquier subconjunto $S \subseteq \{1,\ldots,M\}$ de $s \leq t$ columnas satisface

$$\sum_{\mathbf{a}\in\mathbb{F}_2^s} |n^{-1}\nu_S(\mathbf{a};A) - 2^{-s}| \leq \varepsilon.$$

Para nuestros propósitos, el concepto más importante que necesitaremos será el de conjunto (M,t)-universal.

Definición 8: Un *conjunto (M,t)-universal* B es un subconjunto de $\mathbb{F}_2^M$ tal que para cualquier subconjunto $S \subseteq \{1,\ldots,M\}$ de t posiciones, el conjunto de las proyecciones de los elementos de B en las posiciones S contiene todos los vectores $\mathbf{a} \in \mathbb{F}_2^t$.

Dada una (n,M)-matriz binaria A, si para cualquier subconjunto $S \subseteq \{1,\ldots,M\}$ de t columnas y cualquier vector $\mathbf{a} \in \mathbb{F}_2^s$, se satisface $\nu_S(\mathbf{a};A) > 0$, entonces las filas de A forman un conjunto (M,t)-universal. Nos interesan conjuntos universales del mínimo tamaño posible.

En [16] se establece la conexión entre conjuntos universales y matrices cuasi independientes.

Proposición 9: Sea A una (n,M)-matriz binaria. Para $\varepsilon \leq 2^{-t}$, si A es ε-cuasi t-independiente, entonces las filas de A forman un conjunto (M,t)-universal de tamaño n.

Además, en [19], [20], [16] también se relacionan estos conceptos con matrices de bajo sesgo.

Corolario 10: Sea A una (n,M)-matriz binaria. Si A es ε-sesgada, entonces A es $2^{t/2}\varepsilon$-cuasi t-independiente.

Por tanto, la construcción de conjuntos universales se reduce a la construcción de matrices cuasi independientes mediante la Proposición 9, que a su vez se reduce a la construcción de matrices de bajo sesgo, mediante el Corolario 10. Más adelante, presentaremos una construcción de matrices cuasi independientes aún más eficiente que la aplicación directa del Corolario 10.

III. Construcciones

En esta sección presentamos nuestras construcciones de códigos cuasi SFP. Antes de entrar en detalles explícitos, daremos un razonamiento intuitivo de nuestra propuesta.

No es difícil ver que, en un (n, M)-código aleatorio binario, la probabilidad de que dos coaliciones de tamaño c esten separadas se maximiza cuando se generan las palabras código según un vector de probabilidad $\mathbf{p} = (p_1, \ldots, p_n)$ con $p_1 = \cdots = p_n = 1/2$. Es decir, generamos al azar M palabras código $(u_1, \ldots, u_n)$ tales que $\Pr\{u_i = 1\} = p_i = 1/2$. Pero ya que estamos interesados en códigos ε-cuasi c-SFP, se permitirá un pequeño sesgo en estas probabilidades y, por tanto, consideraremos matrices de bajo sesgo.

Existen construcciones explícitas de (n, M)-matrices ε-sesgadas con $n = 2^{O(\log M + \log \varepsilon^{-1})}$ [16], que conducen a obtener conjuntos (M, t)-universales, de tamaño $2^{O(t)} \log M$. Si disponemos los vectores de este conjunto universal como las filas de una matriz, las columnas de esta matriz formarán un código c-SFP para $t = 2c$. Este código tendrá tamaño M, longitud $2^{O(2c)} \log M$ y tasa $2^{-O(2c)}$. Nuestra idea consist en relajar la propiedad de conjunto universal y permitir que un determinado número de vectores $\mathbf{a} \in \mathbb{F}_2^t$ no aparezcan en cada proyección de t posiciones del conjunto. Esto da lugar a conjuntos "cuasi universales". Finalmente demostramos que los conjuntos "cuasi universales" se pueden utilizar para generar códigos ε-cuasi c-SFP.

III-A. Conjuntos universales y cuasi universales

Los conjuntos universales se han descrito en la Definición 8, y mostrado que la construcción de conjuntos universales se puede reducir a la construcción de matrices ε-sesgadas.

Es fácil ver que un conjunto $(M, 2c)$-universal de tamaño n genera un (n, M)-código (c, c)-separable, es decir, c-SFP [21]. Para ello, sea A una (n, M)-matriz cuyas filas forman un conjunto $(M, 2c)$-universal, y tomemos las columnas de A como las palabras de un código C. Consideremos dos c-coaliciones disjuntas $U, V \subseteq C$, es decir, $2c$ columnas de A. Debido a que las filas de A forman un conjunto $(M, 2c)$-universal, entonces para las $2c$ columnas seleccionadas aparezcan todos los posibles vectores $\mathbf{a} \in \mathbb{F}_2^{2c}$. En particular, hay una fila i donde todas las columnas correspondientes a U contienen el símbolo 0 y todas las columnas correspondientes a V contienen el símbolo 1. Por tanto, i es una posición separadora para las coaliciones U, V. Es decir, $P_i(U) \cap P_i(V) = \emptyset$, como se deseaba.

Construcciones eficientes de conjuntos $(M, 2c)$-universales usando matrices ε-cuasi t-independiente se presentan en [16], en virtud de la Proposición 9 y el Corolario 10. Estas construcciones dan lugar a códigos c-SFP de longitud $2^{O(c)} \log M$. Utilizando esta idea, nuestro objetivo es relajar la restricción que la $(M, 2c)$-universalidad impone para asi obtener una matriz más corta, es decir, un código de mejor tasa. De hecho, no es necesario que todos los posibles vectores de $\mathbb{F}_2^{2c}$ aparezcan en el código. Por lo tanto, nos proponemos relajar la Definición 8 permitiendo que un número máximo de vectores $\mathbf{a} \in \mathbb{F}_2^{2c}$, por ejemplo z, no aparezcan en la proyección de cualquier subconjunto $S \subseteq \{1, \ldots, M\}$ de $2c$ posiciones. Esto se formaliza en la siguiente definición.

Definición 11: Un conjunto (M, t, z)-universal B es un subconjunto de $\mathbb{F}_2^M$ tal que para cada subconjunto $S \subseteq \{1, \ldots, M\}$ de t posiciones el conjunto de proyecciones de los elementos de B sobre los índices de S contiene todos los vectores $\mathbf{a} \in \mathbb{F}_2^t$ excepto, como máximo, z.

De nuevo, si A es una (n, M)-matriz, las filas de A generan un conjunto (M, t, z)-universal siempre que existan al menos $2^t - z$ vectores $\mathbf{a} \in \mathbb{F}_2^t$ tales que $\nu_S(\mathbf{a}; A) > 0$, para todos los subconjuntos $S \subseteq \{1, \ldots, M\}$ de t columnas.

De manera similar a la Proposición 9, el siguiente resultado muestra la conexión entre conjuntos (M, t, z)-universales y matrices ε-cuasi t-independientes.

Proposición 12: Sea A una matriz binaria (n, M). Para $\varepsilon \leq (z+1)2^{-t}$, si A es ε-cuasi t-independiente, entonces las filas de A generan un conjunto (M, t, z)-universal de tamaño n.

Demostración: Por contradicción, supondremos que las filas de A no generan generan un conjunto (M, t, z)-universal. En otras palabras, existe un subconjunto $S \subseteq \{1, \ldots, M\}$ de t columnas tales que hay más de z vectores $\mathbf{a} \in \mathbb{F}_2^t$ tal que $\nu_S(\mathbf{a}; A) = 0$. Para este subconjunto particular S se tiene que

$$\sum_{\mathbf{a} \in \mathbb{F}_2^t} |n^{-1} \nu_S(\mathbf{a}; A) - 2^{-t}| \geq (z+1)2^{-t} + \sum_{\substack{\mathbf{a} \in \mathbb{F}_2^t \text{ t.q.} \\ \nu_S(\mathbf{a}; A) > 0}} |n^{-1} \nu_S(\mathbf{a}; A) - 2^{-t}| \geq (z+1)2^{-t+1} > \varepsilon,$$

que contradice el hecho que la matriz A sea ε-cuasi t-independiente. ■

III-B. Conjuntos (M, t, z)-universales

Según la Proposición 12, la construcción de conjuntos (M, t, z)-universales se reduce a construir una matriz binaria $(z+1)2^{-t}$-cuasi t-independiente, y según el Corolario 10, esto se reduce a la construcción de una matriz ε-sesgada. En realidad, la matriz A del Corolario 10 se puede tomar como una matriz (ε, t)-sesgada, que es una condición menos restrictiva que una matriz ε-sesgada.

En [16] se puede encontrar una construcción estándar para matrices binarias (ε, t)-sesgadas.

Teorema 13: Sea A una (n, M')-matriz binaria ε-sesgada, y sea H la matriz de paridad de un código binario lineal de longitud M, dimensión $M - M'$ y distancia mínima $t + 1$. Entonces, el producto matricial $A \times H$ es una(n, M)-matriz binaria (ε, t)-sesgada.

Habitualmente, la matriz H utilizada en el Teorema 13 es la matriz de paridad de un código BCH binario. En este caso, la matriz H tendrá M columnas y $M' = t \log M$ filas. Entonces, empleando el Teorema 13 en el Corolario 10, el número de filas de una (n, M)-matriz binaria ε-cuasi t-independiente se puede reducir de $n = 2^{O(t + \log M + \log \varepsilon^{-1})}$ a $n = 2^{O(t + \log\log M + \log \varepsilon^{-1})}$ [16].

El problema ahora se reduce a obtener (n, M')-matrices binarias ε-sesgadas con el menor número posible de filas. En [17] se presentan construcciones explicitas de dichas matrices tales que su número de filas es

$$n \leq 2^{2(\log_2 M' + \log_2 \varepsilon^{-1})}.$$

Tabla I
TASAS DE CÓDIGO ALCANZABLES PARA CONSTRUCCIONES EXPLÍCITAS DE CÓDIGOS ε-CUASI c-SFP DE TAMAÑOS ENTRE 10^3 Y 10^7

			Tamaño del código				
c	z	$\log_2 \varepsilon$	$M = 10^3$	$M = 10^4$	$M = 10^5$	$M = 10^6$	$M = 10^7$
2	0	n/a	$1{,}531 \times 10^{-6}$	$1{,}148 \times 10^{-6}$	$9{,}187 \times 10^{-7}$	$7{,}656 \times 10^{-7}$	$6{,}562 \times 10^{-7}$
2	1	n/a	$6{,}124 \times 10^{-6}$	$4{,}593 \times 10^{-6}$	$3{,}675 \times 10^{-6}$	$3{,}062 \times 10^{-6}$	$2{,}625 \times 10^{-6}$
2	2	$-5{,}336 \times 10^{5}$	$1{,}378 \times 10^{-5}$	$1{,}034 \times 10^{-5}$	$8{,}268 \times 10^{-6}$	$6{,}890 \times 10^{-6}$	$5{,}906 \times 10^{-6}$
2	3	$-3{,}001 \times 10^{5}$	$2{,}450 \times 10^{-5}$	$1{,}837 \times 10^{-5}$	$1{,}470 \times 10^{-5}$	$1{,}225 \times 10^{-5}$	$1{,}050 \times 10^{-5}$
3	0	n/a	$1{,}063 \times 10^{-8}$	$7{,}975 \times 10^{-9}$	$6{,}380 \times 10^{-9}$	$5{,}316 \times 10^{-9}$	$4{,}557 \times 10^{-9}$
3	1	n/a	$4{,}253 \times 10^{-8}$	$3{,}190 \times 10^{-8}$	$2{,}552 \times 10^{-8}$	$2{,}127 \times 10^{-8}$	$1{,}823 \times 10^{-8}$
3	2	$-3{,}567 \times 10^{7}$	$9{,}569 \times 10^{-8}$	$7{,}177 \times 10^{-8}$	$5{,}742 \times 10^{-8}$	$4{,}785 \times 10^{-8}$	$4{,}101 \times 10^{-8}$
3	3	$-2{,}006 \times 10^{7}$	$1{,}701 \times 10^{-7}$	$1{,}276 \times 10^{-7}$	$1{,}021 \times 10^{-7}$	$8{,}506 \times 10^{-8}$	$7{,}291 \times 10^{-8}$
3	5	$-8{,}917 \times 10^{6}$	$3{,}828 \times 10^{-7}$	$2{,}871 \times 10^{-7}$	$2{,}297 \times 10^{-7}$	$1{,}914 \times 10^{-7}$	$1{,}640 \times 10^{-7}$
3	6	$-6{,}551 \times 10^{6}$	$5{,}210 \times 10^{-7}$	$3{,}908 \times 10^{-7}$	$3{,}126 \times 10^{-7}$	$2{,}605 \times 10^{-7}$	$2{,}233 \times 10^{-7}$

En [15, Teorema 10] se presenta una mejor construcción explícita con un valor inferior de n. Lamentablemente, las condiciones requeridas en esa construcción hacen que no sea aplicable a nuestro caso, por lo que tenemos que recurrir a la construcción de [17].

A continuación, resumimos los pasos para la construcción explícita de un conjunto (M, t, z)-universal.

1. Tomar $\varepsilon = (z+1)2^{-3t/2}$.
2. Construir una (n, M')-matriz A' ε-sesgada, donde $M' = t \log M$.
3. Construir la matriz de paridad H de un código BCH binario de longitud M, codimensión $M' = t \log M$ y distancia mínima $t+1$.
4. El producto matricial $A = A' \times H$ genera una (n, M)-matriz binaria (ε, t)-sesgada.
5. La matriz A es también ε'-cuasi t-independiente, con $\varepsilon' = 2^{t/2}\varepsilon = (z+1)2^{-t}$. Por tanto, las filas de A generan un conjunto (M, t, z)-universal.

Empleando la construcción de (n, M')-matrices binarias ε-sesgadas proporcionada en [17], el conjunto (M, t, z)-universal resultante tendrá tamaño

$$n = 2^{2(3\,t/2 + \log_2 t + \log_2 \log_2 M - \log_2(z+1))}.$$

III-C. *Códigos cuasi SFP*

Se ha visto que un conjunto $(M, 2c)$-universal de tamaño n genera un (n, M)-código c-SFP. Consideremos una (n, M)-matriz binaria A cuyas filas generan un conjunto $(M, 2c, z)$-universal B, y tomemos las columnas de A como las palabras de un código ε-cuasi c-SFP C, y por tanto tendrá tasa $R = \log M/n$. Para $z < 2^c$ el conjunto $(M, 2c, z)$-universal B es, de hecho, un conjunto (M, c)-universal. Para ver esto, nótese que si un vector de $\mathbb{F}_2^c$ no apareciese en una proyección de c posiciones de B, significaría que faltan $\geq 2^c$ vectores de $\mathbb{F}_2^{2c}$ en alguna proyección de $2c$ columnas. Esto contradice la definición de conjunto $(M, 2c, z)$-universal con $z < 2^c$.

Dado un código C construido usando un conjunto $(M, 2c, z)$-universal, para facilitar el análisis, supondremos que por cada c-coalición $U \subseteq C$, cada posible vector de $\mathbb{F}_2^c$ aparece aproximadamente con probabilidad uniforme (ya que el conjunto $(M, 2c, z)$-universal ha sido generado a partir de una matriz cuasi independiente).

El siguiente corolario formaliza la relación entre códigos ε-cuasi c-SFP y conjuntos (M, t, z)-universales.

Corolario 14: Sean $M > 0$, $c \geq 2$, $z < 2^c$, y $\varepsilon \geq p(M, c, z)$, donde

$$p(M, c, z) \stackrel{\text{def}}{=} M^c (1 - 2^{-c})^n.$$

Entonces, un conjunto $(M, 2c, z)$-universal de tamaño n genera un código ε-cuasi c-SFP de tasa $R = \log M/n$.

Demostración: Considérese un código C generado a partir de un conjunto $(M, 2c, z)$-universal. Sea $\mathbf{z}$ un descendiente generado por una c-coalición del código, $\mathbf{z} \subseteq \text{desc}_c(C)$. Por las suposiciones anteriores, la probabilidad que $\mathbf{z}$ pertenezca a otra c-coalición V es $(1 - 2^{-c})^n$. Por tanto, usando la desigualdad de Boole, se puede acotar la probabilidad de que $\mathbf{z}$ sea generada por otra coalición del código como

$$p(M, c, z) = M^c (1 - 2^{-c})^n.$$

El cociente (probabilidad) de descendientes que no son unívocamente c-decodificables en $\text{desc}_c(C)$ es por tanto $\leq p(M, c, z)$, lo que significa que C es un código ε-cuasi c-SFP. ■

III-D. *Resultados*

En la Tabla I se muestran las tasas obtenidas de códigos cuasi c-SFP para el caso de coaliciones de tamaños $c = 2$ y 3. En número máximo de configuraciones $\{0, 1\}^{2c}$ que faltan se denota como z, y la probabilidad que un descendiente no sea unívocamente c-decodificable se denota mediante ε. Obsérvese que cuando $z < 2$ el código es c-SFP, es decir $\varepsilon = 0$. El valor de ε dado para una fila corresponde al peor caso. Los valores de tasa que se obtienen son del orden de, aproximadamente, 10 veces el valor de la tasa que se obtendría para construcciones explícitas de códigos SFP ordinarios.

IV. APLICACIÓN A CÓDIGOS DE FINGERPRINTING

En esta sección se muestra cómo un código binario ε-cuasi c-SFP se puede utilizar para construir una familia de códigos de fingerprinting equipados con un algoritmo de decodificación eficiente.

Para que un esquema de fingerprinting tenga una probabilidad de error baja, un solo código no es suficiente y se necesita una *familia de códigos* $\{C_j\}_{j \in T}$, siendo T un

conjunto finito. La familia $\{C_j\}_{j\in T}$ es pública. El distribuidor elige un código C_j con probabilidad $\pi(j)$. Esta elección se mantiene en secreto.

En [14, Corolario 1] se proponen condiciones de existencia de una familia de códigos de fingerprinting concatenados, que usan un código cuasi separable como código interno. Obsérvese que el código cuasi separable puede ser sustituido por un código cuasi SFP. Combinando este hecho con los resultados presentados en este trabajo, obtenemos una construcción explícita de un código de fingerprinting.

Corolario 15: Sea $C_{\text{out}} \subseteq \mathbb{F}_q^n$ un código de Reed-Solomon extendido de tasa

$$R_o = R(C_{\text{out}}) < \frac{1-\sigma}{c(c+1)},$$

y sea C_{in} un (l, q)-código ε-cuasi c-SFP de tasa $R_i = R(C_{\text{in}})$, con $\varepsilon < \sigma$. Entonces, existe una construcción explícita de una família de códigos binarios de fingerprinting $\{C_j\}_{j\in T}$ con código externo C_{out} y código interno C_{in}, con un algoritmo de identificación en tiempo polinómico, tasa $R = R_i R_o$ y probabilidad de error decreciendo exponencialmente como

$$p_e \leq 2^{-n\, l(\frac{1-\sigma}{c} R_i - (c+1)R + o(1))} + 2^{-nD(\sigma\|\varepsilon)}.$$

Por último, vale la pena señalar aquí que, como se muestra en [12], [14], el uso de códigos ε-cuasi c-SFP en lugar de códigos SFP ordinarios introduce un término de error adicional en el proceso de identificación, como se indica en el Corolario 15. Afortunadamente, este término de error disminuye exponencialmente con la longitud del código exterior.

V. Conclusiones

Los códigos cuasi separables y cuasi SFP son dos versiones menos restrictivas de los códigos separables. En este trabajo, hemos presentado las primeras construcciones explícitas de códigos cuasi SFP.

Nuestro trabajo parte del estudio de la conexión entre matrices de bajo sesgo y conjuntos universales, y la posterior conexión entre conjuntos universales y códigos separables.

A partir de esta idea, hemos introducido una relajación en la definición de conjunto universal. Se demuestra que un conjunto cuasi universal se puede utilizar para construir un código cuasi SFP. Esta observación nos ha llevado a las construcciones explícitas de códigos cuasi SFP.

También hemos demostrado cómo las construcciones propuestas pueden ser usadas para construir de forma explícita una familia de códigos concatenados de fingerprinting. La construcción presentada se basa en los resultados teóricos de existencia de un trabajo anterior, que presupone la existencia de códigos cuasi SFP. Por lo tanto, una de las principales aportaciones de este trabajo ha sido la de proporcionar una implementación "verdadera" de de dicha existencia teórica de un esquema de fingerprinting.

Por último, cabe señalar que a pesar de que un conjunto universal genera un código separable, la relación entre un conjunto cuasi universal y un código cuasi separable no es en absoluto evidente y será objeto de investigación futura.

Agradecimientos

J. Moreira y M. Fernández han sido financiados por el Gobierno de España mediante los proyectos CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES" y TEC2011-26491 "COPPI", y por la Generalitat de Catalunya mediante la ayuda 2009 SGR-1362.

G. Kabatiansky ha sido financiado por la Russian Foundation for Basic Research mediante las ayudas RFBR 13-07-00978 y RFBR 13-01-12458.

Referencias

[1] D. R. Stinson, T. van Trung, and R. Wei, "Secure frameproof codes, key distribution patterns, group testing algorithms and related structures," *J. Stat. Plan. Infer.*, vol. 86, no. 2, pp. 595–617, May 2000.

[2] J. N. Staddon, D. R. Stinson, and R. Wei, "Combinatorial properties of frameproof and traceability codes," *IEEE Trans. Inf. Theory*, vol. 47, no. 3, pp. 1042–1049, Mar. 2001.

[3] D. Tonien and R. Safavi-Naini, "Explicit construction of secure frameproof codes," *Int. J. Pure Appl. Math.*, vol. 6, no. 3, pp. 343–360, 2003.

[4] D. R. Stinson and G. M. Zaverucha, "Some improved bounds for secure frameproof codes and related separating hash families," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2508–2514, June 2008.

[5] A. D. Friedman, R. L. Graham, and J. D. Ullman, "Universal single transition time asynchronous state assignments," *IEEE Trans. Comput.*, vol. C-18, no. 6, pp. 541–547, June 1969.

[6] Y. L. Sagalovich, "Separating systems," *Probl. Inform. Transm.*, vol. 30, no. 2, pp. 105–123, 1994.

[7] M. S. Pinsker and Y. L. Sagalovich, "Lower bound on the cardinality of code of automata's states," *Probl. Inform. Transm.*, vol. 8, no. 3, pp. 59–66, 1972.

[8] Y. L. Sagalovich, "Completely separating systems," *Probl. Inform. Transm.*, vol. 18, no. 2, pp. 140–146, 1982.

[9] J. Körner and G. Simonyi, "Separating partition systems and locally different sequences," *SIAM J. Discr. Math. (SIDMA)*, vol. 1, no. 3, pp. 355–359, Aug. 1988.

[10] G. D. Cohen and H. G. Schaathun, "Asymptotic overview on separating codes," Deptartment of Informatics, University of Bergen, Norway, Tech. Rep. 248, Aug. 2003.

[11] G. D. Cohen and H. G. Schaathun, "Upper bounds on separating codes," *IEEE Trans. Inf. Theory*, vol. 50, no. 6, pp. 1291–1294, June 2004.

[12] M. Fernández, G. Kabatiansky, and J. Moreira, "Almost separating and almost secure frameproof codes," in *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, Saint Petersburg, Russia, Aug. 2011, pp. 2696–2700.

[13] A. Barg, G. R. Blakley, and G. Kabatiansky, "Digital fingerprinting codes: Problem statements, constructions, identification of traitors," *IEEE Trans. Inf. Theory*, vol. 49, no. 4, pp. 852–865, Apr. 2003.

[14] J. Moreira, G. Kabatiansky, and M. Fernández, "Lower bounds on almost-separating binary codes," in *Proc. IEEE Int. Workshop Inform. Forensics, Security (WIFS)*, Foz do Iguaçu, Brazil, Nov. 2011, pp. 1–6.

[15] J. Bierbrauer and H. Schellwat, "Almost independent and weakly biased arrays: Efficient constructions and cryptologic applications," in *Proc. Int. Cryptol. Conf. (CRYPTO)*, ser. Lecture Notes Comput. Sci. (LNCS), vol. 1880, Santa Barbara, CA, Aug. 2000, pp. 533–544.

[16] J. Naor and M. Naor, "Small-bias probability spaces: Efficient constructions and applications," *SIAM J. Comput. (SICOMP)*, vol. 22, no. 4, pp. 838–856, Aug. 1993.

[17] N. Alon, O. Goldreich, J. Håstad, and R. Peralta, "Simple constructions of almost k-wise independent random variables," *Random Struct. Alg.*, vol. 3, no. 3, pp. 289–304, 1992.

[18] D. Boneh and J. Shaw, "Collusion-secure fingerprinting for digital data," *IEEE Trans. Inf. Theory*, vol. 44, no. 5, pp. 1897–1905, Sept. 1998.

[19] U. V. Vazirani, "Randomness, adversaries and computation," Ph.D. dissertation, Dept. Elect. Eng. Comp. Sci., Univ. California, Berkeley, 1986.

[20] P. Diaconis, *Group Representations in Probability and Statistics*. Beachwood, OH: Inst. Math. Stat., 1988.

[21] N. Alon, V. Guruswami, T. Kaufman, and M. Sudan, "Guessing secrets efficiently via list decoding," *ACM Trans. Alg.*, vol. 3, no. 4, pp. 1–16, Nov. 2007.

Mejorando la seguridad de un criptosistema OPE mediante la uniformización de los datos

Santi Martínez*, Daniel Sadornil†, Josep Conde*, Magda Valls* y Rosana Tomàs*
* Departament de Matemàtica, Universitat de Lleida
Email: {santi,jconde,magda,rosana}@matematica.udl.cat
† Departamento de Matemáticas, Estadística y Computación, Universidad de Cantabria
Email: sadornild@unican.es

***Resumen*—La cantidad de información almacenada en bases de datos crece constantemente. Una base de datos contiene múltiples registros divididos en varios campos. Algunos de éstos pueden contener información sensible, así que es necesario evitar que se acceda a ellos. Tradicionalmente, para proteger este tipo de información, se hace uso de la criptografía, pero la criptografía convencional tiene el problema de que, para consultas que necesitan acceder a un campo para todos los registros, se requiere descifrar el campo entero.**

La criptografía ordenable asegura que comparar datos cifrados produce el mismo resultado que comparar los datos originales, lo que permite ordenarlos sin descifrarlos. Por tanto, con este sistema se admiten búsquedas y consultas por intervalos en campos cifrados.

En este artículo proponemos un complemento, compatible con múltiples criptosistemas ordenables, consistente en transformar los datos de manera que se oculte su distribución estadística como paso previo al cifrado ordenable.

***Palabras clave*—bases de datos *(databases)*, criptografía ordenable *(order preserving encryption)*, criptografía simétrica *(symmetric key cryptography)*, distribución de probabilidad *(probability distribution)*, privacidad *(privacy)*, uniformización de datos *(data uniformization)*.**

I. Introducción

La criptografía permite esconder información sensible ante atacantes potenciales. Sin embargo, la información cifrada puede tener diferentes usos que requieran técnicas criptográficas específicas. En particular, este trabajo se centra en la privacidad y/o seguridad en bases de datos.

La seguridad en bases de datos es esencial para evitar un acceso no autorizado a información sensible. A título de ejemplo, en *The Toronto Star* [1] se explicó como un banco vendió un disco duro en eBay, olvidándose de borrar los datos en claro de centenares de clientes.

En bases de datos seguras, a veces se necesita permitir ciertas operaciones, o consultas, que requerirían que se descifrasen todos los campos necesarios para realizarlas, e.g. obtener los registros con un campo entre dos valores.

La criptografía ordenable (OPE, *Order Preserving Encryption*) permite hacer comparaciones de orden con datos cifrados, pues garantiza que éstos conservan el orden establecido entre datos en claro. Así, si un campo está cifrado de esta manera, las consultas de rango se pueden realizar eficientemente y asegurando que un atacante que tuviera acceso a la información almacenada en la base de datos no pueda obtener información de los datos en claro.

Consideremos una base de datos médica cifrada y supongamos que queremos saber el número de pacientes en un grupo de edad. Si el criptosistema usado no conserva el orden, esa consulta requerirá que cifremos cada uno de los valores del intervalo de edad y los comparemos con el campo correspondiente, o, alternativamente, que descifremos el campo de edad de todos los registros y los comparemos con los límites del intervalo. Si el algoritmo de cifrado no es determinista (o si en lugar de un entero el campo contiene un número real, como el nivel de azúcar en sangre) sólo la segunda alternativa es válida.

En cambio, si la base de datos usa OPE, sólo necesitamos cifrar los extremos del intervalo y comprobar cuantos registros tienen su campo cifrado de edad entre esos dos valores.

En esencia, un esquema OPE es una función estrictamente creciente del conjunto de datos en claro al conjunto de datos cifrados. Su seguridad recae en que dicha función, aun manteniendo el orden, parezca lo más aleatoria posible [2]. Esto asegurará que sólo aquellos con un conocimiento exacto de como calcularla (lo que está determinado por la clave del criptosistema) serán capaces de invertirla.

Los esquemas OPE son necesariamente simétricos, puesto que el conocimiento de la función de cifrado permite aproximar, hasta cualquier precisión, la función de descifrado. Nótese que, si un atacante con capacidad de cifrar valores arbitrarios quiere descifrar un valor concreto, podría realizar una búsqueda dicotómica de dicho valor, hasta llegar a una aproximación satisfactoria. Así, un hipotético OPE asimétrico sería automáticamente vulnerable a este ataque.

Los esquemas OPE han sido diseñados para entornos en que exista la posibilidad de que un intruso pueda acceder a la base de datos cifrada pero que no pueda cifrar ni descifrar valores arbitrarios.

Desde el punto de vista de un atacante, saber que un campo concreto ha sido cifrado con un esquema OPE proporciona una fuente útil de información si puede acceder a los datos almacenados. Si conociera los valores en claro de un conjunto de valores cifrados, podría crear una aproximación de la función de descifrado. Por lo que, si accediera a nuevos valores cifrados, podría usarla para aproximar los valores en claro correspondientes.

Por ejemplo, un atacante conoce x_1, x_2, y_1, y_2 y que $y_1 = Enc(x_1)$ y $y_2 = Enc(x_2)$; si obtuviera un valor cifrado y, con

$y_1 < y < y_2$, entonces sabría que su descifrado, x, pertenece al intervalo (x_1, x_2). Además, podría deducir un valor x' cuya proximidad a x dependería de la predictibilidad de la función y la distancia entre los valores que supiera de antemano.

Por tanto, una función OPE debería minimizar este problema asegurando un alto nivel de impredecibilidad siendo lo más aleatoria posible [2]. Además, es deseable ocultar la distribución estadística de los datos cifrados.

En este artículo, proponemos la transformación de los datos a cifrar con un esquema OPE de manera que la entrada al criptosistema acabe siendo lo más uniforme posible. Consideramos un esquema OPE que cifra datos pertenecientes al intervalo $[0, 1]$, por lo que, antes de cifrar, convertiremos los datos desde la distribución inicial a una uniforme en $[0, 1]$. De forma similar, tras el descifrado del esquema OPE, convertiremos el valor obtenido, a un valor de la distribución inicial.

El resto del artículo está estructurado en las siguientes secciones: La sección II, expone algunas de las propuestas anteriores. En la sección III se propone la forma de escoger la distribución más apropiada para los datos. La sección IV explica cómo se complementa un sistema OPE concreto y como afecta la transformación a su eficiencia. Finalmente, las conclusiones se dan en la sección V.

II. Trabajos previos

Durante la última década, debido, en parte, al aumento de la preocupación por la privacidad de los datos preservando el análisis de los mismos, la criptografía ordenable ha experimentado un gran interés.

Bebek, en [3], realizó una primera propuesta donde proponía un método para cifrar un entero p añadiendo los p primeros valores de una secuencia pseudo-aleatoria segura de enteros positivos. Sin embargo, el coste de cifrar un valor p de n bits mediante este método es exponencial en n. Además, si μ es la media de la distribución de la secuencia pseudo-aleatoria, entonces $f(x) = \mu x$ aproxima la función de cifrado, y $f^{-1}(x) = x/\mu$, la de descifrado. Estas aproximaciones serán menos útiles si μ es cercano a 0 y la distribución tiene una desviación grande.

En [4], Ozsoyoglu et al. propusieron el uso de polinomios para el cifrado de enteros. Dichos polinomios no deben tener ningún extremo en el intervalo al que pertenecen los datos. Pero, el hecho de que algunos polinomios no dispongan de una fórmula explícita de su inversa, lleva a los autores a proponer la composición de varios polinomios fácilmente invertibles, de manera que el descifrado consista en aplicar las inversas en orden inverso. Para evitar desbordamientos de enteros, deciden controlar los coeficientes y usar logaritmos, lo que requerirá tratar con números en coma flotante y errores de precisión. Esto hace que la elección de la clave sea un proceso complejo. Además, descifrar es mucho más difícil que cifrar.

Agrawal et al. [5] propusieron la transformación de datos que siguieran cierta distribución estadística en datos que mantuvieran el orden y siguieran una distribución distinta, escogida de antemano. Para generar la función de cifrado, hacen uso de todos los datos a cifrar disponibles, así como una muestra de valores de la distribución objetivo. Por tanto, el tiempo de generación de la clave es lineal en el tamaño de la base de datos. Al cifrar, los datos se transforman en una distribución uniforme y, desde ésta, en la distribución objetivo. Para hacer esto, separan los datos en diversas particiones y, dentro de éstas, usan interpolación lineal. Si después de haber generado la clave se añadieran una gran cantidad de datos a la base de datos, podría ser necesario escoger una nueva clave y volver a cifrar la base de datos.

En [6], Lee et al. propusieron el esquema COPE (*Chaotic Order Preserving Encryption*). En este esquema, que reparte los datos en subconjuntos, se altera el orden de los mismos en función de la clave, por lo que no es un esquema OPE puro. El hecho de que haya que reordenar los subconjuntos para responder a una consulta afecta negativamente al coste.

Boldyreva et al. [2] presentaron una función OPE basada en el uso de un algoritmo de muestreo para la distribución hipergeométrica. Señalan el hecho de que, para funciones OPE con datos en claro y cifrados tomando valores enteros, el conjunto de salida es mayor que el de entrada (lo que permite que no haya dos valores en claro a los que corresponda el mismo valor cifrado). Así, una función de $\{1, ..., M\}$ a $\{1, ..., N\}$, con $M < N$, puede ser determinada unívocamente mediante la elección de un subconjunto (de cardinal M) del conjunto de salida que contendrá el cifrado de los valores del conjunto de entrada. Es más, basándose en las diferentes formas de hacer esta elección, proponen un criterio que debe cumplir una buena función OPE, recordando que, básicamente, la función, aun manteniendo el orden, tiene que ser lo más aleatoria posible.

En [7], se propuso un método OPE que cifraba datos pertenecientes al intervalo real $[0, 1]$. La función de cifrado se obtenía como composición de varias funciones básicas y el descifrado consistía en componer sus inversas en orden inverso. Cada función básica se definía por dos segmentos, el primero, desde el origen hasta el punto $P_{k_i} = (x_{k_i}, y_{k_i})$, y el segundo, de P_{k_i} al punto $(1, 1)$. La colección de todos los P_{k_i} constituía la clave del criptosistema. Para evitar funciones de mala calidad, la región donde se escogían los puntos de la clave estaba acotada.

En [8], se exponen dos metodologías para analizar la calidad de una función OPE. La primera de ellas se basa en la conversión de la función de cifrado en una secuencia que poder analizar como una señal de ruido. La segunda se basa en calcular las diferencias entre la función de cifrado y las aproximaciones que un atacante puede calcular a partir de un pequeño conjunto de puntos conocidos.

Más recientemente, se ha desarrollado un nuevo esquema OPE [9], que construye de forma iterativa una serie de puntos por los que pasará la función de cifrado. Inicialmente, la función es la identidad de $[0, 1]$ a $[0, 1]$. Así, en el primer nivel, se considera el rectángulo, cuyos lados son paralelos a los ejes, y con esquinas en $(0, 0)$ y $(1, 1)$, y se elige un punto en la diagonal descendiente. Este punto permite definir dos nuevos rectángulos con un vértice en el nuevo punto y otro en uno de los puntos extremos iniciales. En un segundo

nivel, se elige un punto en la diagonal descendiente de cada uno de los dos rectángulos, lo que permite definir cuatro rectángulos más pequeños. El proceso se repite hasta obtener la precisión deseada. Los puntos obtenidos constituyen la clave del criptosistema.

En este artículo, proponemos complementar un sistema OPE mediante la transformación de los datos a cifrar, de manera que oculte mejor la distribución estadística que éstos tuvieran inicialmente. El objetivo es que, antes de aplicar la función OPE, los datos sigan una distribución uniforme en $[0, 1]$.

III. Modelos de distribución de probabilidad de los datos

Nuestra propuesta consiste en complementar un sistema OPE mediante la ocultación de la distribución de probabilidad de los datos a cifrar. Por tanto, la transformación que proponemos no pretende servir como método de cifrado en sí mismo, sino como un preproceso que mejore la seguridad del sistema completo (para que los datos cifrados no conserven las propiedades de la distribución inicial).

Dicha ocultación de la distribución se realizará mediante la función de distribución acumulada (CDF, *Cumulative Distribution Function*). De esta forma, si X es una variable aleatoria continua con CDF F_X, entonces la variable aleatoria $P = F_X(X)$ tiene distribución uniforme en $[0, 1]$. Así, para recuperar la distribución original se usará la función cuantil F_X^{-1} (inversa de la CDF). En realidad, la transformación no necesita conocer la distribución real de los datos, pues es suficiente utilizar una que tenga el mismo soporte (i.e. que el intervalo para el que la función de densidad es no nula coincida en ambos casos). Esto se debe a que, para una variable X' con el mismo soporte que X, la variable $P' = F_X(X')$ también está distribuida entre 0 y 1, aunque ya no siga una distribución uniforme. Evidentemente, si conocemos la distribución de los datos a cifrar, lo ideal es utilizar la CDF y la cuantil de dicha distribución, pues es lo que dará mejor resultado.

Dependiendo del rango de valores que puedan tener los datos a cifrar consideramos tres grandes familias de distribuciones: con soporte finito, soporte infinito acotado inferiormente (o superiormente), y soporte infinito no acotado. Dentro de cada familia escogemos una distribución que, asumiendo un mínimo de información adicional, maximice la entropía.

Según el principio de máxima entropía, la distribución de probabilidad que mejor representa a una variable aleatoria es aquella en que, dadas unas ciertas condiciones, la desinformación es máxima. Así, una distribución de máxima entropía es aquella en que su entropía es al menos tan grande como la de cualquier otra distribución de su clase (donde una *clase* es el conjunto de distribuciones que cumplen una serie de restricciones). Por tanto, si de una distribución sólo se conocen unos pocos parámetros, la distribución a escoger es la que tenga máxima entropía para esos parámetros, lo que asegura que la distribución asume el mínimo de información adicional. Además, muchos sistemas tienden a seguir distribuciones de este tipo de manera natural.

En función del tipo de soporte hemos considerado las siguientes clases:

- Distribuciones con un mínimo y un máximo que conocemos. La distribución de máxima entropía para esta clase es la distribución uniforme continua.
- Distribuciones con mínimo y media conocidos. La distribución de máxima entropía para esta clase es la distribución exponencial.
- Distribuciones con media y varianza conocidas. La distribución de máxima entropía para esta clase es la distribución normal.

En la tabla I se muestran la CDF y la función cuantil de las distribuciones consideradas. Éstas son: la uniforme $\mathcal{U}(a, b)$, donde a y b son el mínimo y el máximo del soporte; la exponencial $\mathcal{E}(\lambda, \theta)$, donde θ es el mínimo, $\lambda = (\mu - \theta)^{-1}$ y μ es la media; y la normal $\mathcal{N}(\mu, \sigma^2)$, donde μ es la media y σ^2 la varianza.

Si tratamos con datos acotados sólo superiormente, cambiando el signo a cada valor, es posible usar también con ellos la distribución exponencial. Una vez obtenido su valor uniformizado, habrá que reflejar el resultado para no invertir el orden de los datos, i.e. $p = 1 - F_{\mathcal{E}(\lambda,\theta)}(-x)$.

III-A. *Determinación de la distribución*

Para poder determinar la distribución que se aproxima más adecuadamente a los datos a cifrar, necesitamos conocer información de los mismos. Pues, en función de que estén o no acotados inferiormente y/o superiormente, usaremos una distribución u otra. Si no conocemos esa información, pero tenemos una muestra con algunos de los datos a cifrar, trataremos de inferir la distribución a partir de éstos.

De hecho, si no se dispone de ninguna información de los datos a cifrar y ni siquiera tenemos una muestra que analizar, la opción más sensata es utilizar una distribución normal $\mathcal{N}(0, 1)$. Con ello nos aseguramos que cualquier valor que nos encontremos se transforme en un valor entre 0 y 1.

En lo sucesivo, se asume que disponemos únicamente de una muestra S con n valores $(x_1, x_2, \ldots x_n)$ de los datos a cifrar (donde puede haber valores repetidos). Con esta muestra, deberemos determinar cual de las distribuciones consideradas es la más adecuada.

En esta sección proponemos un método sencillo, basado en la cantidad de datos que hay en tres subintervalos iguales entre el mínimo y el máximo. Existen métodos alternativos, desde dividir el rango en una cantidad mayor de subintervalos (y proceder de manera similar), hasta estimar primero los parámetros de las distribuciones y luego hacer una prueba χ^2 de Pearson [10] para ver cuál se ajusta mejor.

Para el método de los tres subintervalos, primero buscamos el *mínimo*, m, y el *máximo*, M, de la muestra que tenemos. Ambos valores existirán siempre, incluso aunque la distribución no esté acotada, ya que S es finito. A continuación, partimos el intervalo $[m, M]$ en tres partes iguales, mediante los valores $l_1 = m + \frac{1}{3}(M - m)$ y $l_2 = m + \frac{2}{3}(M - m)$ y contamos cuantos datos hay en cada uno de los subintervalos:

Tabla I
FUNCIONES DE DISTRIBUCIÓN ACUMULADA Y CUANTIL

Distribución	CDF	Cuantil
$\mathcal{U}(a,b)$	$p = \begin{cases} 0 & \text{para } x < a \\ \frac{x-a}{b-a} & \text{para } a \le x < b \\ 1 & \text{para } x \ge b \end{cases}$	$x = a + p(b-a)$ para $0 \le p \le 1$
$\mathcal{E}(\lambda,\theta)$	$p = \begin{cases} 0 & \text{para } x < \theta \\ 1 - e^{-\lambda(x-\theta)} & \text{para } x \ge \theta \end{cases}$	$x = \begin{cases} \theta - \frac{\ln(1-p)}{\lambda} & \text{para } 0 \le p < 1 \\ +\infty & \text{para } p = 1 \end{cases}$
$\mathcal{N}(\mu,\sigma^2)$	$p = \frac{1}{2}\left[1 + \operatorname{erf}\left(\frac{x-\mu}{\sqrt{2\sigma^2}}\right)\right]$	$x = \begin{cases} -\infty & \text{para } p = 0 \\ \mu + \sqrt{2\sigma^2}\,\operatorname{erf}^{-1}(2p-1) & \text{para } 0 < p < 1 \\ +\infty & \text{para } p = 1 \end{cases}$

$c_1 = \#\{x \in S | x < l_1\}$, $c_2 = \#\{x \in S | l_1 \le x \le l_2\}$, $c_3 = \#\{x \in S | x > l_2\}$.

Nótese que, si la distribución sólo está acotada inferiormente, el máximo (cuando n crezca) tenderá a infinito, por lo que es razonable suponer que los datos se acumularán en el primer subintervalo (o en el tercero, si está acotada sólo superiormente). Asimismo, si la distribución no está acotada, podemos asumir que los datos tenderán a acumularse en el segundo subintervalo. En cambio, si el soporte de la distribución es finito, no se espera que los datos tiendan a acumularse tanto en ningún subintervalo.

Por tanto, si en el primer subintervalo hay tantos datos como en los otros dos juntos ($c_1 \ge c_2 + c_3$) usaremos la CDF de la exponencial, $p = F_{\mathcal{E}(\lambda,\theta)}(x)$, si es el central el que tiene tantos como los otros dos ($c_2 \ge c_1 + c_3$) usaremos la CDF de la normal, $p = F_{\mathcal{N}(\mu,\sigma^2)}(x)$, y, si es el tercero ($c_3 \ge c_1 + c_2$) usaremos también la exponencial con los cambios necesarios ($p = 1 - F_{\mathcal{E}(\lambda,\theta)}(-x)$). Finalmente, si ningún subintervalo domina a los otros dos, usaremos la CDF de la uniforme, $p = F_{\mathcal{U}(a,b)}(x)$.

Una vez escogida la distribución más adecuada para modelar los datos, deberemos estimar sus parámetros, lo que será diferente en cada caso particular.

III-B. Estimación de los parámetros

En esta sección proponemos los métodos de estimación de los parámetros de cada una de las distribuciones consideradas.

Tal como se ha indicado para la determinación de la distribución, si conocemos los valores exactos de los parámetros que queremos estimar, los usaremos directamente. Si no es el caso, podemos utilizar la muestra para tratar de hacer una estimación.

De entre los diversos tipos de estimadores se ha optado por utilizar estimadores insesgados [11], [12] de mínima varianza (UMVUE, *Uniformly Minimum-Variance Unbiased Estimator*). Sin embargo, para los parámetros que afectan al soporte de la distribución se ha optado por realizar algunos ajustes que disminuyan la posibilidad de encontrarnos con valores fuera de rango.

A continuación exponemos los estimadores que se han usado para las distinta distribuciones.

III-B1. Distribución uniforme: Ésta requiere dos parámetros a y b que son los valores extremos del soporte.

Notemos que todo valor menor o igual que $\hat{a}$ (la estimación de a) será interpretado como $\hat{a}$ y todo valor mayor o igual que $\hat{b}$ será interpretado como $\hat{b}$, por tanto, si el estimador de a tiene sesgo, es preferible que sea hacia la izquierda, i.e. $\hat{a} \le a$. De la misma manera, es preferible que $\hat{b} \ge b$.

Para determinar los valores $\hat{a}$ y $\hat{b}$ partiremos de los estimadores UMVUE, que son $\hat{a}' = m - \frac{M-m}{n-1}$ y $\hat{b}' = M + \frac{M-m}{n-1}$. Estos estimadores son insesgados, pero puede suceder que $\hat{a}'$ sea mayor que a o $\hat{b}'$ sea menor que b, por lo que les aplicaremos unas correcciones para disminuir dicha posibilidad.

En primer lugar, la media μ de la distribución debería coincidir con el valor medio de a y b. Por tanto, si la media muestral $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ no coincide con $\mu' = \frac{1}{2}(\hat{a}' + \hat{b}')$, modificaremos uno de los dos parámetros para forzar la coincidencia. Si $\bar{x} < \mu'$, disminuiremos el estimador de a: $\hat{a}'' = 2\bar{x} - \hat{b}'$, manteniendo $\hat{b}'' = \hat{b}'$. Si $\bar{x} > \mu'$, aumentaremos el estimador de b: $\hat{b}'' = 2\bar{x} - \hat{a}'$, manteniendo $\hat{a}'' = \hat{a}'$.

En segundo lugar, la varianza σ^2 de la distribución debería coincidir con $\frac{1}{12}(b-a)^2$. Si la varianza muestral $s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$ es mayor que ${\sigma^2}'' = \frac{1}{12}(\hat{b}'' - \hat{a}'')^2$, modificaremos ambos parámetros para forzar que coincidan. Para ello, ampliaremos el soporte en ambas direcciones de manera que no se altere el punto medio del intervalo, así, $\hat{a} = \hat{a}'' - \delta$ y $\hat{b} = \hat{b}'' + \delta$, donde $\delta = s\sqrt{3} - \frac{1}{2}(\hat{b}'' - \hat{a}'')$. Nótese que, si $s^2 < {\sigma^2}''$ no haremos esta última modificación, pues nos obligaría a reducir el soporte (lo que no es deseable), en tal caso, los estimadores serían $\hat{a} = \hat{a}''$ y $\hat{b} = \hat{b}''$.

III-B2. Distribución exponencial: Para determinar la distribución exponencial adecuada se requiere el valor de los parámetros λ y θ.

Para λ nos basaremos en que su valor es el inverso de la desviación estándar σ, i.e. $\lambda = \sigma^{-1}$. Por ello, usaremos un estimador UMVUE de este parámetro $\hat{\sigma} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - m)$ y luego lo invertiremos, $\hat{\lambda} = \hat{\sigma}^{-1}$. Como este parámetro no afecta al soporte de la distribución no haremos ninguna corrección adicional.

Para el parámetro θ, que corresponde a la cota inferior del soporte, es deseable que $\hat{\theta} \le \theta$, pues todo valor menor o igual que $\hat{\theta}$ será interpretado como $\hat{\theta}$. Por ello, escogeremos el menor de dos estimadores, el primero de ellos el UMVUE, que es $\hat{\theta}' = m - \hat{\sigma}/n$, y el segundo $\hat{\theta}'' = \bar{x} - \hat{\sigma}$ (que se basa en que la media μ de la distribución debería coincidir con $\theta + \sigma$). Así, $\hat{\theta} = \min(\hat{\theta}', \hat{\theta}'')$.

III-B3. Distribución normal: Esta distribución requiere los parámetros μ, la media, y σ^2, la varianza.

Ninguno de los dos parámetros afecta al soporte de la distribución, pues éste es infinito no acotado, por tanto, podemos utilizar los estimadores UMVUE en ambos casos.

Así, para μ usaremos la media muestral: $\hat{\mu} = \bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$. Y, para σ^2, la varianza muestral aplicando la corrección de Bessel: $\hat{\sigma^2} = s^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2$.

Nótese que la corrección de Bessel se necesita porque para el cálculo de la varianza muestral usamos la media muestral $\bar{x}$. De hecho, si conociéramos la media poblacional μ (pero no la varianza), calcularíamos la varianza muestral usando μ (en vez de $\bar{x}$), en cuyo caso no usaríamos la corrección de Bessel, i.e. $\hat{\sigma^2} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$.

IV. Complementando un criptosistema concreto

El complemento propuesto en este artículo puede adaptarse a múltiples esquemas OPE. Con el fin de evaluar el sobrecoste que conlleva en el tiempo de cifrado y descifrado hemos decidido adaptarlo al criptosistema DOPE (*Diagonal-based Order Preserving Encryption*) [9] y realizar una serie de experimentos cuyos resultados mostramos en la tabla II.

El esquema DOPE cifra datos en $[0, 1]$, por lo que la adaptación es inmediata. Antes de cifrar, convertiremos los datos iniciales utilizando la función CDF de la distribución escogida mediante los criterios de la sección III. El resultado, que es un valor entre 0 y 1, se cifra mediante el esquema DOPE. Similarmente, después de descifrar, el valor obtenido se convierte a la distribución original mediante la función cuantil de la distribución correspondiente (ver tabla I).

Para la implementación se ha utilizado el criptosistema DOPE con pequeñas variaciones de seguridad y formato en los datos. Estos cambios han hecho que los tiempos de cifrado y descifrado sean mayores que los presentados en el artículo original.

Como se vio en las fórmulas de la tabla I, la conversión para la distribución uniforme es trivial. En el caso de la distribución exponencial, antes de cifrar habrá que calcular una exponenciación y, después de descifrar, un logaritmo. Asimismo, antes de cifrar un dato que siga una distribución normal habrá que calcular la función error complementaria erfc (pues la conversión es más rápida con ésta que con otras implementaciones de la CDF). Para cada una de estas tres funciones, se ha usado la implementación de la biblioteca estándar de C para valores de tipo `long double`. La función cuantil de la normal se ha implementado mediante el método de Newton-Raphson [13], por lo que consta de un bucle en el que, en cada iteración, se calculan la CDF (que, a su vez, calcula la función erfc) y la función de densidad (que requiere una exponenciación). Se ha escogido este método por ser de convergencia cuadrática.

Fijado un nivel de seguridad l, la clave del criptosistema DOPE es una lista de 2^l puntos (por los que pasa la función de cifrado). El cifrado (y el descifrado) requieren una búsqueda dicotómica para localizar la posición del valor a cifrar (o a descifrar) en la lista de abscisas (u ordenadas), por lo que su coste temporal es logarítmico en el tamaño de la lista, o, lo que es lo mismo, lineal en l.

La tabla II muestra los resultados para el criptosistema DOPE básico (sin cambio de distribución) y usando cada una de las tres distribuciones contempladas. Se ha usado un ordenador portátil con 2,4 GHz de CPU, 4 GB de RAM y el sistema operativo Debian GNU/Linux.

Se han realizado pruebas con un nivel de seguridad $14 < l < 19$, generando diez claves para cada valor de l. Con cada una de las claves se han cifrado y descifrado un millón de valores sin cambio de distribución, otro millón en el que los datos a cifrar seguían una distribución uniforme $\mathcal{U}$, otro más en que seguían una exponencial $\mathcal{E}$ y, finalmente, un millón más de datos que seguían una normal $\mathcal{N}$.

En la tabla II, las columnas T. cif. y T. des. indican el tiempo necesario para cifrar o descifrar un valor sin realizar ningún cambio de distribución. Las columnas T. cif. $\mathcal{U}$, $\mathcal{E}$ y $\mathcal{N}$ muestran el tiempo necesario para cifrar un valor que siga una de esas tres distribuciones, por lo que antes de usar la función de cifrado del esquema DOPE se debe uniformizar el valor mediante la CDF de la distribución. Las columnas T. des. $\mathcal{U}$, $\mathcal{E}$ y $\mathcal{N}$ muestran el tiempo necesario para descifrar un valor y luego convertirlo a su distribución original mediante la función cuantil de la distribución.

Podemos observar que el sobrecoste que conlleva cifrar o descifrar datos que siguen una distribución uniforme es de unos 10 ns y que éste es ligeramente mayor en el caso del descifrado. Si los datos siguen una exponencial, ambos sobrecostes son inferiores a 100 ns, siendo el del cifrado mayor que el del descifrado. Finalmente, en el caso de la normal, el sobrecoste al cifrado también es inferior a los 100 ns (incluso menor que el sobrecoste de la exponencial), pero para el descifrado, la implementación de la función cuantil ha causado un sobrecoste de más de 800 ns (y difícilmente podría ser mucho menor sin reducir la precisión).

Como es lógico, el sobrecoste debido al cambio de distribución es independiente del nivel de seguridad del esquema DOPE (y, por tanto, del tamaño de la clave), pues conlleva un proceso previo al cifrado o posterior al descifrado. Esto implica que, si se complementa cualquier otro criptosistema OPE, se deberían obtener sobrecostes similares.

V. Conclusión

En este artículo se ha propuesto un método de uniformización de datos como paso previo a la función de cifrado de un criptosistema ordenable. La finalidad de esta transformación es ocultar la distribución de probabilidad que siguen los datos en claro. Antes de cifrar datos que sigan una distribución conocida usaremos su función de distribución acumulada para convertirlos a una distribución uniforme en $[0, 1]$, y similarmente, después de descifrar usaremos la función cuantil para recuperar la distribución original.

También se ha propuesto un método para asignar una distribución de probabilidad a datos cuya distribución real no sea conocida. Se han considerado tres posibles casos: si los datos aparentan estar acotados en ambas direcciones,

Tabla II
RESULTADOS DE LA EXPERIMENTACIÓN

l	**T. cif.**	**T. des.**	**T. cif.** $\mathcal{U}$	**T. des.** $\mathcal{U}$	**T. cif.** $\mathcal{E}$	**T. des.** $\mathcal{E}$	**T. cif.** $\mathcal{N}$	**T. des.** $\mathcal{N}$
14	122,4 ns	122,7 ns	137,7 ns	135,1 ns	214,2 ns	195,6 ns	204,3 ns	976,2 ns
15	133,2 ns	136,2 ns	147,5 ns	148,9 ns	222,1 ns	207,0 ns	212,4 ns	1003,1 ns
16	195,1 ns	194,7 ns	204,4 ns	205,9 ns	285,2 ns	260,9 ns	265,9 ns	1074,3 ns
17	215,9 ns	212,7 ns	221,1 ns	223,3 ns	303,9 ns	285,1 ns	282,9 ns	1084,5 ns
18	236,9 ns	236,6 ns	244,5 ns	249,9 ns	326,1 ns	301,6 ns	307,1 ns	1122,9 ns
19	253,3 ns	253,4 ns	262,3 ns	275,0 ns	349,5 ns	328,7 ns	327,7 ns	1147,9 ns

se tratarán como si siguieran una distribución uniforme; si aparentan estar acotados sólo por un lado, se tratarán como si siguieran una exponencial (si la cota es superior requerirá una pequeña modificación); y, si aparentan no estar acotados, se tratarán como si siguieran una distribución normal.

Se han escogido estas distribuciones porque con ellas se pueden tratar datos definidos sobre cualquier tipo de intervalo, sea o no infinito. Además de eso, la uniforme es la distribución de máxima entropía de entre todas las que tienen soporte finito, la exponencial lo es entre todas las que tienen al menos una cota y tienen la misma media y la normal es la distribución de máxima entropía de entre todas las que tienen misma media y varianza.

La elección de la distribución y la estimación de sus parámetros se han realizado con el objetivo de disminuir la posibilidad de encontrarnos con valores fuera de su soporte. Nótese que, si el error supone asignar una distribución con soporte infinito a unos datos acotados o realizar una estimación de un máximo superior al real, dicho error, aunque no es deseable, no causará problemas.

Los métodos propuestos se han implementado y se ha comprobado qué sobrecoste representan en el tiempo de cifrado y descifrado de un criptosistema OPE.

AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por los proyectos MTM2010-21580-C02-01/02 y MTM2010-16051 del Gobierno de España y SGR2014-1666 de la Generalitat de Catalunya.

REFERENCIAS

[1] T. Hamilton, "Error sends bank files to eBay," The Toronto Star, September 15, 2003.

[2] A. Boldyreva, N. Chenette, Y. Lee, and A. O'neill, "Order-Preserving Symmetric Encryption," *Advances in Cryptology - EUROCRYPT 2009*, vol. LNCS, no. 5479, pp. 224–241, 2009, ISSN 0302-9743.

[3] G. Bebek, "Anti-tamper database research: Inference control techniques," Case Western Reserve University, Technical Report EECS 433, 2002, Final Report.

[4] G. Ozsoyoglu, D. A. Singer, and S. S. Chung, "Anti-Tamper Databases: Querying Encrypted Databases," in *17th Annual IFIP WG 11.3 Working Conference on Database and Applications Security*, 2003.

[5] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu, "Order Preserving Encryption for Numeric Data," in *ACM SIGMOD international conference on Management of data*, 2004, pp. 563–574.

[6] S. Lee, T.-J. Park, D. Lee, T. Nam, and S. Kim, "Chaotic Order Preserving Encryption for Efficient and Secure Queries on Databases," *IEICE Transactions on Information and Systems*, vol. E92-D, no. 11, pp. 2207–2217, 2009.

[7] S. Martínez, V. Mateu, R. Tomàs, and M. Valls, "Criptografía ordenable para bases de datos," in *XII Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*, U. Zurutuza, R. Uribeetxeberria, and I. Arenaza-Nuño, Eds. Donostia–San Sebastián, España: Mondragon Unibertsitatea, September 2012, pp. 35–40, ISBN 978-84-615-9933-2.

[8] S. Martínez, J. M. Miret, R. Tomàs, and M. Valls, "Security Analysis of Order Preserving Symmetric Cryptography," *Applied Mathematics & Information Sciences (AMIS)*, vol. 7, no. 4, pp. 1285–1295, July 2013, ISSN 1935-0090.

[9] S. Martínez, J. M. Miret, R. Tomàs, and M. Valls, "Securing Databases by using Diagonal-based Order Preserving Symmetric Encryption," *Applied Mathematics & Information Sciences (AMIS)*, vol. 8, no. 5, pp. 2085–2094, September 2014, ISSN 1935-0090.

[10] K. Pearson, "On the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling," *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, vol. 50, no. 302, pp. 157–175, 1900.

[11] V. G. Voinov and M. S. Nikulin, *Unbiased Estimators and Their Applications*, ser. Mathematics and Its Applications. Dordrecht, Netherlands: Kluwer Academic Publishers, 1993, vol. 1: Univariate Case.

[12] K. Lam, B. K. Sinha, and Z. Wu, "Estimation of parameters in a two-parameter exponential distribution using ranked set sample," *Annals of the Institute of Statistical Mathematics*, vol. 46, no. 4, pp. 723–736, 1994.

[13] J. Raphson, *Analysis Aequationum Universalis seu Ad Aequationes Algebraicas Resolvendas Methodus Generalis, & Expedita, Ex nova Infinitarum Serierum Methodo, Deducta ac Demonstrata*. London: Th. Braddyll, 1690.

Análisis e Implementación del Generador SNOW 3G Utilizado en las Comunicaciones 4G

J. Molina-Gil, P. Caballero-Gil
Departamento de Informática
Universidad de La Laguna
Email: jmmolina@ull.edu.es, pcaballe@ull.edu.es

A. Fúster-Sabater
Instituto de Física Aplicada.
Consejo Superior de Investigaciones Científicas
Email: amparo@iec.csic.es

***Resumen*—La cuarta generación de telefonía móvil, comercializada como tecnología 4G, y conocida como LTE (Long Term Evolution) o su versión mejorada LTE-A (LTE-Advanced), se está implantando con rapidez por todo el mundo. Teniendo en cuenta su carácter inalámbrico y móvil, la seguridad de estas comunicaciones resulta crucial. En este trabajo se presenta un estudio teórico y un análisis práctico del generador SNOW 3G, que constituye el núcleo de la integridad y la confidencialidad de las comunicaciones 4G. El objetivo es evaluar la implementación y funcionamiento de este generador en dispositivos móviles con el fin de proponer mejoras, principalmente en cuanto a eficiencia.**

***Palabras clave*—Criptografía de clave secreta (*Secret key cryptography*), Generador pseudoaleatorio (*Pseudorandom generator*), Cifrado en flujo (*Stream cipher*), SNOW 3G, 4G, LTE, LTE-A.**

I. Introducción

El aumento en el consumo de datos móviles, y la aparición de numerosas aplicaciones y servicios de banda ancha son, junto a debilidades detectadas en la seguridad de las comunicaciones, los principales motivos que han conducido a la progresiva sustitución de la tecnología 3G o UMTS por la nueva tecnología 4G, conocida como LTE o LTE-A. El completo despliegue de las redes LTE comerciales en España es inminente, encontrándose actualmente en funcionamiento solo en algunas grandes ciudades. En cuanto a otros países, ya se ha implantado la tecnología 4G en: cuatro países de África, once de América, diecinueve de Asia, veintinueve de Europa, y dos de Oceanía.

En general, la evolución de cualquier sistema de telecomunicaciones suele implicar la mejora de sus características de seguridad gracias al aprendizaje a partir de las debilidades y los ataques sufridos por sus predecesores. Concretamente, la evolución de los sistemas de cifrado utilizados para proteger la confidencialidad en telefonía móvil se puede resumir de la forma siguiente. En primer lugar, el cifrado en flujo A5/1 y su versión A5/2 fueron utilizados en el estándar de telefonía 2G o GSM. En ambos cifrados se detectaron serias debilidades, razón por la cual el sistema de cifrado en bloque conocido como Kasumi los sustituyó en la tecnología 3G o UMTS. En 2010, el cifrado Kasumi fue atacado y roto con recursos computacionales muy modestos y en consecuencia, el sistema de cifrado tuvo que ser modificado de nuevo para la tecnología del nuevo estándar 4G o LTE, de manera que fue el cifrado en flujo SNOW 3G el que se propuso para la protección de la confidencialidad e integridad de las comunicaciones.

El objetivo de este trabajo es realizar un análisis del generador SNOW 3G, que constituye el núcleo tanto del algoritmo de confidencialidad UEA2 en LTE (EEA1 en LTE-A), como del de integridad UIA2 en LTE (EIA1 en LTE-A), [1]. Este generador proporciona una gran velocidad en la generación de datos, lo que lo hace en general muy apropiado para su uso en dispositivos con recursos limitados.

El contenido de este trabajo se organiza del siguiente modo. En la Sección II se incluye un resumen de trabajos relacionados con la temática. Los principales conceptos y notaciones utilizados se introducen en la Sección III. La Sección IV presenta algunos detalles de la implementación llevada acabo en la plataforma iOS, así como una evaluación de su rendimiento. Finalmente, la Sección V cierra este estudio con algunas conclusiones y trabajos futuros.

II. Estado del Arte

Los predecesores del SNOW 3G fueron el SNOW 1.0 [2] y el SNOW 2.0 [3]. La versión original, SNOW 1.0, fue presentada al proyecto NESSIE, pero pronto se detectaron algunas debilidades y se lanzaron diversos ataques. Uno de ellos [4], con complejidad computacional de $O(2^{224})$, se basó en la posibilidad de recuperar la clave con solo conocer una salida del generador de longitud 2^{95}. Otro criptoanálisis con similar complejidad computacional fue el ataque por diferenciación (distinguishing attack) [5], que también requería de una salida de tamaño 2^{95} .

Estos y otros ataques demostraron la existencia de algunas debilidades en el diseño del generador SNOW 1.0, por lo que surgió una nueva versión, más segura, denominada SNOW 2.0. Este es hoy en día uno de los dos cifrados en flujo elegidos para el estándar ISO/IEC IS 18033-4 [6]. Este generador se basa en unos principios de diseño similares a los del cifrado en bloque SOSEMANUK, que es uno de los cuatro cifrados finalistas dentro del perfil software seleccionados para el eSTREAM Portfolio [7].

Posteriormente, durante su evaluación por el European Telecommunications Standards Institute (ETSI), el diseño del SNOW 2.0 fue modificado para aumentar su resistencia a ataque algebraicos [8] de forma que dicha modificación dio lugar al SNOW 3G. El ETSI publicó un informe técnico acerca de su diseño [9], pero hasta el momento no se ha hecho pública ninguna evaluación completa del diseño del SNOW 3G.

En las publicaciones existentes, el SNOW 3G se ha revelado como un generador con una gran resistencia a ataques por diferenciación lineal [10] [11], pero débil frente a otro tipos de ataques. Uno de los primeros y más sencillos intentos de criptoanálisis fue el ataque propuesto en [12]. Otro ataque [13], basado en la sincronización de la caché y en datos de tiempo empíricos, permitió recuperar el estado inicial en cuestión de segundos y sin necesidad de conocer ningún bit. Este tipo de ataque se basa en el hecho de que operaciones como las permutaciones y multiplicaciones por la constante α y su inversa, son implementadas realmente utilizando tablas de búsqueda. El trabajo presentado en [14] describe un estudio del mecanismo de resincronización del SNOW 3G usando ataques por colisión, con una complejidad de $O(2^8)$. Finalmente, el generador SNOW 3G ha sido sujeto de otros estudios de complejidad, tales como las publicaciones [15] y [16]. En este trabajo se proporciona un nuevo estudio, enfocado hacia los aspectos más prácticos de su implementación.

III. Descripción Teórica del Generador Snow 3G

Los cifrados en flujo están basados en generadores pseudo-aleatorios cuyos bits de salida son operados con los bits del texto en claro mediante una XOR, para generar bit a bit el texto cifrado. Su principal ventaja es que en general permiten obtener el texto cifrado a una gran velocidad, lo que los hacen especialmente apropiados para comunicaciones que requieran eficiencia, y para dispositivos con recursos limitados, como por ejemplo los teléfonos móviles ya que la comunicación debe ser inmediata y disponen de una batería limitada.

El generador en flujo analizado en este trabajo tiene un estructura típica de generador no lineal basado en un registro de desplazamiento con realimentación lineal o LFSR (Linear Feedback Shift Register).

Los siguientes términos y notación se utilizan en este documento para describir la estructura e implementación del generador SNOW 3G:

$GF(2) = \{0, 1\}$ Cuerpo de Galois con dos elementos, 0 y 1.

$GF(2)[x]$ Anillo de polinomios en una indeterminada x, con coeficientes en $GF(2)$.

$p(x)$ Polinomio primitivo en $GF(2)[x]$.

d Grado de un polinomio $p(x)$.

$GF(2^d)$ Cuerpo extendido de $GF(2)$, definido por un polinomio $p(x)$ de grado d, y con 2^d elementos.

$GF(2^d)[x]$ Anillo de polinomios en una indeterminada x con coeficientes en $GF(2^d)$.

$\beta \in GF(2^8)$ Raíz del polinomio $x^8 + x^7 + x^5 + x^3 + 1$ perteneciente a $GF(2)[x]$.

$\alpha \in GF(2^{32})$ Raíz del polinomio $x^4 + \beta^{23}x^3 + \beta^{245}x^2 + \beta^{48}x + \beta^{239}$ perteneciente a $GF(2^8)[x]$.

s_t estado de 32-bits perteneciente a un LFSR.

$=$ Operador de asignación.

$\oplus$ Operación XOR bit a bit.

$\boxplus$ Suma de enteros módulo 2^{32}.

$||$ Concatenación de dos operandos.

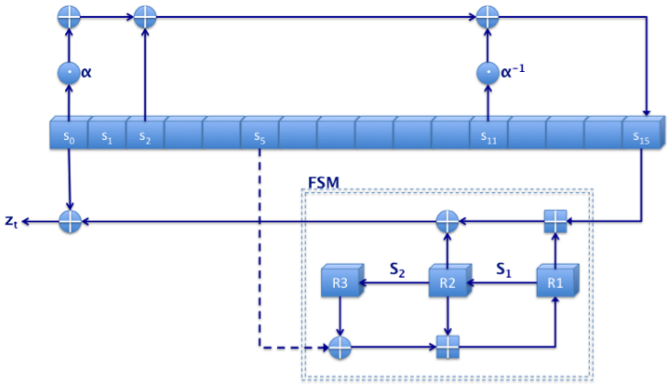


Figura 1. Generador SNOW 3G

La figura 1 muestra la estructura del generador SNOW 3G. Como puede verse, consta de dos partes principales: un LFSR y una máquina de estados finitos o FSM (Finite State Machine).

El LFSR consta de 16 estados $s_0, s_1, s_2, ..., s_{15}$, de forma que cada uno de ellos contiene 32 bits. La función de realimentación se define mediante un polinomio primitivo definido sobre un cuerpo extendido $GF(2^{32})$, e implica dos multiplicaciones, una por una constante $\alpha \in GF(2^{32})$ y otra por la inversa de α. A continuación se muestra la expresión de dicha realimentación:

$$s_{t+16} = \alpha s_t \oplus s_{t+2} \oplus \alpha^{-1} s_{t+11}, \forall t \geq 0 \qquad (1)$$

La FSM constituye la parte no lineal del generador y se alimenta de dos valores de entrada procedentes del LFSR correspondientes a los estados s_5 y s_{15}. La FSM está formada por tres registros de 32 bits, R1, R2 y R3, y dos cajas de sustitución o S-boxes, S1 y S2, que se utilizan para actualizar los registros R2 y R3. Las dos S-Boxes S1 y S2 mapean los 32 bits de entrada a 32 bits de salida mediante la aplicación de varias combinaciones de una S-box básica sobre cada uno de los 4 bytes de entrada. La estructura de la caja S1 se basa en la S-box utilizada en el cifrado estándar AES (Advanced Encryption Standard), mientras que la S-box S2 fue especialmente diseñada para el SNOW 3G. Por último están las operaciones de mezcla que se utilizan en la FSM, donde se realiza una XOR bit a bit, y una suma de enteros módulo 2^{32}.

El LFSR base del generador SNOW 3G puede usarse en dos modos de operación diferentes: de inicialización y de generación de secuencia cifrante. Por una parte, cuando opera en modo de inicialización, el generador se desplaza en cada pulso de reloj sin producir ninguna salida. Por otra parte, en modo de generación, tras cada pulso de reloj el generador se desplaza y produce una salida de una palabra de 32 bits. De hecho, se puede decir que el SNOW 3G es un generador orientado a palabras ya que produce una secuencia de salida de 32 en 32 bits, bajo el control de una clave de 128 bits y un vector de inicialización o IV (Initialization Vector) de 128 bits.

Con respecto a la implementación del SNOW 3G, que es el principal objeto de estudio de este trabajo, se pueden hacer varias observaciones.

En primer lugar, las dos multiplicaciones implicadas en el LFSR pueden ser implementadas como desplazamientos de bytes con una XOR con alguno de los 2^8 patrones posibles, tal como se muestra a continuación. Dado que β es la raíz del polinomio primitivo $x^8 + x^7 + x^5 + x^3 + 1$, el cuerpo extendido $GF(2^8)$ puede ser generado a partir de sucesivas potencias de β. Por tanto, el conjunto $\{0, 1, \beta, \beta^2, \beta^3, ..., \beta^{2^8-2}\}$ representa todo el cuerpo extendido $GF(2^8)$. De ahí tenemos que cualquier elemento de $GF(2^8)$ puede ser representado también mediante un polinomio en $GF(2)[x]$ de grado menor que 8, o bien con un byte cuyos bits se corresponden con los coeficientes de dicho polinomio. De esa manera, las operaciones en $GF(2^8)$ se corresponden con operaciones módulo el polinomio $x^8+x^7+x^5+x^3+1$. Esto significa que en particular, la multiplicación de dos elementos en $GF(2^8)$ resulta de la multiplicación de los dos polinomios correspondientes, posteriormente dividida por el polinomio $x^8 + x^7 + x^5 + x^3 + 1$ de manera que el resto es la salida resultante. La implementación de esta operación como una multiplicación binaria se describe a continuación. Teniendo en cuenta los dos bytes a multiplicar, para cada bit igual a uno en uno de los multiplicandos, se realizan varios desplazamientos a izquierda en el otro byte a multiplicar. Además, cada vez que el bit más a la izquierda del byte original antes del desplazamiento es 1, se realiza una XOR bit a bit con $A9_{16} = 10101001_2$, que es el byte correspondiente al polinomio $x^8+x^7+x^5+x^3+1$. El número de desplazamientos a izquierda a realizar viene dado por la posición de los bits iguales a 1 en el primer multiplicador.

Por otra parte, dado que α es una raíz del polinomio primitivo en $GF(2^8)[x]$, $x^4 + \beta^{23}x^3 + \beta^{245}x^2 + \beta^{48}x + \beta^{239}$, se puede generar el cuerpo extendido $GF(2^{32})$ como sucesivas potencias de α, de forma que el conjunto $\{0, 1, \alpha, \alpha^2, \alpha^3, ..., \alpha^{2^{32}-2}\}$ define todo el cuerpo $GF(2^{32})$. Por tanto se deduce que cualquier elemento de $GF(2^{32})$ puede ser representado mediante un polinomio en $GF(2^8)[x]$ de grado menor que 4, o bien con una palabra de 4 bytes correspondientes a los 4 coeficientes de dicho polinomio. De esa manera, las operaciones en $GF(2^{32})$ corresponden a operaciones con polinomios módulo $x^4 + \beta^{23}x^3 + \beta^{245}x^2 + \beta^{48}x + \beta^{239}$. Esto significa que en particular, la multiplicación de α por cualquier palabra de 4 bytes $(c_3, c_2, c_1, c_0) \in GF(2^{32})$ resulta de la multiplicación de x por el polinomio $c_3x^3 + c_2x^2 + c_1x + c_0$, que es entonces dividida por el polinomio $x^4 + \beta^{23}x^3 + \beta^{245}x^2 + \beta^{48}x + \beta^{239}$, de forma que el resultado obtenido es el resto $(c_2+c_3\beta^{23})x[^3+(c_1+c_3\beta^{245})x^2+(c_0+c_3\beta^{48})x+c_3\beta^{239}$, o lo que es lo mismo, la palabra de 4 bytes $(c_2 + c_3\beta^{23}, c_1 + c_3\beta^{245}, c_0 + c_3\beta^{48}, c_3\beta^{239})$, $\forall c \in GF(2^8)$. De idéntica manera, la multiplicación de α^{-1} por cualquier palabra de 4 bytes $(c_3, c_2, c_1, c_0) \in GF(2^{32})$ resulta de la multiplicación de x^{-1} por el polinomio $c_3x^3+c_2x^2+c_1x+c_0$, que es $c_3x^2 + c_2x + c_1 + c_0x^{-1}$. Como $xx^{-1} = 1$ y $\beta^{255} = 1, x^{-1}$ puede expresarse como $\beta^{255-239}x^3+\beta^{255-239+23}x^2+\beta^{255-239+245}x+\beta^{255-239+48} = \beta^{16}x^3+\beta^{39}x^2+\beta^6x+\beta^{64}$.

Tabla I
DISPOSITIVO UTILIZADO PARA LA EVALUACIÓN

iPhone 3GS			
Arquitectura	**Frecuencia CPU**	**Cache L1I/L1D/L2**	**RAM**
Armv7-A	600 MHz	16Kb/16Kb/256Kb	256MB

Así, la salida resultante del producto es el resto $(c_0\beta^{16})x^3 + (c_3 + c_0\beta^{39})x^2 + (c_2 + c_0\beta^6)x + (c_1 + c_0\beta^{64})$, o lo que es lo mismo, la palabra de 4 bytes $(c_0\beta^{16}, c_3+c_0\beta^{39}, c_2+c_0\beta^6, c_1+c_0\beta^{64})$. En conclusión, una rápida implementación binaria de esta operación puede estar basada en tablas precalculadas de los valores $(c\beta^{16}, c\beta^{39}, c\beta^6, c\beta^{64})$, $\forall c \in GF(2^8)$.

IV. IMPLEMENTACIÓN EN IOS Y EVALUACIÓN

Esta sección recoge un estudio y comparación de diferentes implementaciones software con el fin de concluir cuál es la implementación óptima en dispositivos con recursos limitados, como son los teléfonos móviles. Se ha analizado la implementación del SNOW 3G en plataformas móviles, en concreto para la plataforma iOS, siendo Objective C el lenguaje de programación utilizado. En particular, se han realizado diversos estudios en un iPhone 3GS, cuyas características principales se presentan en la Tabla I.

El primer aspecto a tener en cuenta es que los LFSRs han sido tradicionalmente diseñados para operar sobre el cuerpo de Galois $GF(2)$, lo que es muy apropiado para implementaciones hardware. Sin embargo, cuando se trata de implementaciones software, los LFSRs en los que cada etapa contiene un solo bit tienen una eficiencia menor. Teniendo en cuenta que la mayoría de los microprocesadores en teléfonos móviles tienen una longitud de palabra de 32 bits, la implementación del LFSR del SNOW 3G se espera que sea más eficiente que la de esos otros generadores, ya que en el SNOW 3G el LFSR se define sobre el cuerpo extendido $GF(2^{32})$. Teniendo esto en cuenta, se puede afirmar que el hecho de que la implementación del SNOW 3G se realice teniendo en cuenta el cuerpo $GF(2^{32})$, resultará más adecuado para la arquitectura que actualmente soportan los teléfonos móviles.

El segundo aspecto a considerar está relacionado con las operaciones aritméticas que se deben realizar en el generador, y específicamente, la multiplicación en el cuerpo extendido, porque la función de realimentación en el SNOW 3G implica diversas sumas y multiplicaciones, siendo la multiplicación la operación con mayor coste computacional.

Todos los resultados mostrados fueron obtenidos utilizando Instruments, que es un analizador y visualizador del rendimiento de aplicaciones en código OS X e iOS. Es una herramienta flexible y potente, que permite realizar el seguimiento de uno o más procesos y examinar los datos obtenidos.

Los datos proporcionados corresponden a una media de 10 ejecuciones de las pruebas realizadas, en las cuales se generaron 10^7 bytes de secuencia cifrante utilizando la plataforma mencionada. La Tabla II muestra el tiempo total (en milisegundos) de cada función correspondiente a la implementación del SNOW 3G.

Tabla II
FUNCIONES CON MÉTODO RECURSIVO

Resumen		
Función	**Tiempo (ms)**	**%**
MULxPow	29054,9	92,88
ClockLFSRKeyStreamMode	572	1,77
DIValpha	356,6	1,1
main	264,7	0,8
MULalpha	326,8	0,99
GenerateKeystream	243,8	0,73
ClockFSM	180,3	0,54
S2	128,1	0,34
S1	129,9	0,37
Generator	1,3	0
Total	30258,5	100

Tabla III
FUNCIONES CON TABLAS PRECOMPUTADAS

Coste Computacional		
Función	**Tiempo (ms)**	**%**
ClockLFSRKeyStreamMode	347,3	28,69
main	277,2	22,35
ClockFSM	182,2	14,95
S1	146	12,01
S2	138	11,3
GenerateKeystream	107,3	8,84
Generator	1,3	0,04
Total	1199,4	100

Los resultados muestran claramente que la multiplicación es la función más costosa. La segunda función más costosa es el desplazamiento del LFSR, que se realiza en cada pulso de reloj. A continuación se presentan dos técnicas diferentes para llevar a cabo la multiplicación, así como diferentes formas de implementar el desplazamiento del LFSR propuestas en [11], con el objetivo de descubrir cuál es la implementación óptima.

IV-A. Multiplicación

Tras realizar la implementación del SNOW 3G tal como está propuesta en [1], la Tabla II muestra los resultados de tiempo consumido por cada función. Como puede verse, la función MULxPow utilizada para multiplicar por α y por α^{-1} es la que más tiempo consume.

En la implementación, la multiplicación puede ser programada como una serie de desplazamientos de bytes recursivos y XORs condicionales, o bien como una búsqueda en un tabla con resultados precalculados. Además, en cada pulso de reloj del LFSR, el polinomio de realimentación usa dos funciones MUL_α y DIV_α, que se definen como se muestra a continuación:

$$MUL_\alpha = \\ MUL_xPOW(c, 23, 0xA9)||MUL_xPOW(c, 245, 0xA9) \\ MUL_xPOW(c, 48, 0xA9)||MUL_xPOW(c, 239, 0xA9)$$

$$DIV_\alpha = \\ MUL_xPOW(c, 16, 0xA9)||MUL_xPOW(c, 39, 0xA9) \\ MUL_xPOW(c, 6, 0xA9)||MUL_xPOW(c, 64, 0xA9)$$

La primera propuesta de implementación de la multiplicación resulta más apropiada para sistemas con recursos de memoria limitados, ya que no requiere de espacio de almacenamiento. Sin embargo, como se puede ver en la Tabla II, esta implementación tiene un coste computacional significativo.

La segunda propuesta, que implica el uso de tablas precalculadas, proporciona resultados óptimos, tal como se puede ver en la Tabla III. Tal como muestran los resultados, esta implementación se puede considerar la más rápida para la multiplicación y supone una mejora del 96 % respecto al método recursivo en cuando a tiempo consumido. Sin embargo, uno de los mayores problemas de esta propuesta es la necesidad de almacenamiento, lo que puede ser un problema en dispositivos con recursos limitados. En particular, para SNOW 3G, la tabla consta de 256 elementos de 32 bits cada uno, lo que supone un total de 32*256 bits. Además, la implementación implica dos tablas, una para la función MUL_α y otra para DIV_α lo que significa un total de 2048 bytes. Esta cantidad no supone un gran problema dadas las características de los teléfonos móviles actuales, por lo que por eficiencia, la conclusión obtenida es que este método parece bastante adecuado para los dispositivos analizados.

IV-B. LFSR

Las estructuras de los LFSRs son en general bastante difíciles de implementar en software de forma eficiente. La razón principal es el desplazamiento de los 16 estados en cada pulso de reloj. Este desplazamiento en una implementación hardware, se realiza de manera simultánea por lo que el proceso completo puede ser realizado en un simple pulso de reloj. Sin embargo, en una implementación software, el proceso es iterativo y por lo tanto costoso.

Como se ve en la Tabla III, una vez optimizada la multiplicación, la función ClockLFSRKeyStreamMode es la que más tiempo consume. Con el fin de buscar una mejora de los tiempo consumidos por esta función hemos utilizado diferentes técnicas de optimización descritas en [17], junto con técnicas secuenciales (hardcode) propuestas en la especificación del SNOW 3G.

El método hardcode consiste en la incorporación de los datos directamente en el código fuente en vez de utilizar bucles e índices como hacen el resto de propuestas. El coste de este método de hardcode se corresponde con 15 asignaciones secuenciales. Esta técnica, a pesar de ser más larga, parece que requiere menos tiempo. A continuación se muestra la implementación de este método.

```
void ClockLFSRKeyStreamMode()
 u32 v = (((LFSR_S0 << 8) & 0xffffff00 )^
 (MULalpha((u8)((LFSR_S0>>24) & 0xff))) ^
 (LFSR_S2) ^
 ((LFSR_S11 >> 8) & 0x00ffffff)
```

Tabla IV
EJECUCIÓN CON DIFERENTES MÉTODOS DE IMPLEMENTACIÓN PARA EL LFSR

Funciones	Tradicional	HardCode	Búfer Circular	Ventanas Deslizantes	Desdoblamiento de Bucles
ClockLFSRKeyStreamMode	491,1	342,5	834	184,1	291,3
Generator	1,4	1,3	1,3	1,8	1,1
GenerateKeystream	65,8	65,8	198,3	88,2	68,4
main	246,6	306,8	296,8	297	294,4
Tiempo total	804,9	716,4	1330,4	571,1	655,2

```
(DIValpha((u8)((LFSR_S11) & 0xff)))
);
LFSR_S0 = LFSR_S1;
LFSR_S1 = LFSR_S2;
LFSR_S2 = LFSR_S3;
LFSR_S3 = LFSR_S4;
LFSR_S4 = LFSR_S5;
LFSR_S5 = LFSR_S6;
LFSR_S6 = LFSR_S7;
LFSR_S7 = LFSR_S8;
LFSR_S8 = LFSR_S9;
LFSR_S9 = LFSR_S10;
LFSR_S10 = LFSR_S11;
LFSR_S11 = LFSR_S12;
LFSR_S12 = LFSR_S13;
LFSR_S13 = LFSR_S14;
LFSR_S14 = LFSR_S15;
LFSR_S15 = v;
```

El análisis llevado a cabo incluye una comparación de diferentes técnicas de implementación de un LFSR. Para ello se ha utilizado el método basado en tablas de multiplicaciones precalculadas descrito como óptimo en la sección anterior, y se realiza cada experimento para 10^7 bytes de salida generados por el LFSR del SNOW 3G.

Los valores obtenidos se resumen en la Tabla IV, que muestra el tiempo consumido por las funciones que implican el desplazamiento del LFSR y sus operaciones, así como el tiempo total consumido por cada una de las diferentes propuestas.

Los resultados muestran que el método de hardcode propuesto en la especificación del SNOW 3G no es la mejor opción. Aunque representa un 11 % de mejora sobre el método tradicional, es el método de ventanas deslizantes el que presenta los mejores resultados, con un 29 % de mejora respecto al método tradicional y un 20 % respecto a la propuesta de hardcode. Por otra parte, la peor propuesta es la de búfer circular. Como puede deducirse de los tiempos obtenidos, el método no es aplicable debido a que la actualización de diferentes índices implica el uso de aritmética modular, lo que no resulta muy eficiente con ese método.

Por tanto, se podría concluir que una óptima implementación del LFSR implica el uso de tablas precalculadas para la multiplicación y la técnica de ventanas deslizantes para el desplazamiento del LFSR.

Sin embargo, esta nueva propuesta de implementación del LFSR puede afectar a otra parte del SNOW 3G como puede ser la FSM. Por esta razón, el principal objetivo ahora es determinar si la mejora en la implementación del LFSR puede afectar negativamente a otras partes del código, o bien concluir cuál es la mejora total que se puede llegar a alcanzar.

Tabla V
EJECUCIÓN EN MODO OPTIMIZADO

Coste Computacional		
Función	**Tiempo (ms)**	**%**
ClockLFSRKeyStreamMode	184,7	15,28
main	282,3	22,23
ClockFSM	195,4	17,68
S1	135,9	12,65
S2	163,4	16,33
GenerateKeystream	118,2	10,95
Generator	1,2	1,07
Total	1081,1	100

Para analizarlo, se ha implementado el SNOW 3G con estas dos posibles mejoras. En la Tabla V se muestra el resumen de los resultados obtenidos. Si se comparan los resultados con los de la Tabla III, se puede ver claramente que esta implementación mejora los tiempos para las funciones $ClockLFSRKeyStreamMode, S1$ y $GenerateKey$. Sin embargo, otras funciones como $ClockFSM, S2, GenerateKeystream$ han aumentado ligeramente sus tiempos. La función con el peor resultado es $S2$, que ha incrementado su valor en un 26 % respecto a la propuesta previa. Por otra parte, la mejora más importante ha sido para la función $ClockLFSRKeyStreamMode$, con un 47 %. Todos estos resultados suponen en su conjunto una mejora del 10 % respecto a la propuesta de la especificación.

V. CONCLUSIONES Y TRABAJOS FUTUROS

En este trabajo se ha presentado un análisis, tanto desde un punto de vista teórico como práctico, del generador que se utiliza para la protección de la confidencialidad y la integridad en la generación 4G de telefonía móvil. En particular, después de una introducción teórica del generador SNOW 3G, y varios estudios de implementación del generador en la plataforma móvil iOS, se ha realizado una comparación entre diferentes propuestas, y se han obtenido conclusiones interesantes sobre cómo mejorar la eficiencia de su implementación a través de la optimización del software.

Dado que este es un trabajo en progreso, todavía hay muchos problemas abiertos, tales como el análisis de parámetros no analizados en este trabajo, la utilización de diferentes arquitecturas, así como un estudio comparativo entre ellas. También otro trabajo futuro es la propuesta de una versión ligera del generador SNOW 3G para dispositivos con recursos limitados, y el análisis de propiedades teóricas del generador.

AGRADECIMIENTOS

Investigación financiada por el MINECO y la fundación Europea FEDER mediante los proyectos TIN2011-25452 e IPT-2012-0585-370000.

REFERENCIAS

[1] ETSI/SAGE. Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 & UIA2. Document 2, "SNOW 3G Specification, version 1.1", http://www.3gpp.org/ftp/, 2006.

[2] P. Ekdahl, T. Johansson "SNOW - a new stream cipher", *Proceedings of First Open NESSIE Workshop*, 2000.

[3] P. Ekdahl, T. Johansson "A New Version of the Stream Cipher SNOW", *Proceedings of Selected Areas in Cryptography*, LNCS 2595, pp. 47-61, 2003.

[4] P. Hawkes, G.G. Rose, "Guess-and-determine attacks on SNOW", *Proceedings of Selected Areas in Cryptography*, LNCS 2595, pp. 37-46, 2003.

[5] D. Coppersmith, S. Halevi, C. Jutla, "Cryptanalysis of stream ciphers with linear masking", Proceedings of CRYPTO, LNCS 2442, pp. 515-532, 2002.

[6] ISO/IEC 18033-4:2005. Information technology - Security techniques - Encryption algo-rithms - Part 4: Stream ciphers. http://www.iso.org/iso/home/store/catalogue_ics/

[7] C. Berbain, O. Billet, A. Canteaut, N. Courtois, H. Gilbert, A. Gouget, H. Sibert, "Sosemanuk, a fast software-oriented stream cipher eSTREAM, ECRYPT Stream Cipher", *ECRYPT-Network of Excellence in Cryptology, Call for stream Cipher Primitives-Phase 2*, http://www.ecrypt. eu. org/stream, 2005.

[8] O. Billet, H. Gilbert, "Resistance of SNOW 2.0 Against Algebraic Attacks", *Proceedings of CT-RSA,* LNCS 3376, pp. 19-28, 2005.

[9] ETSI/SAGE Technical report: Specification of the 3GPP Confidentiality and Integrity Al-gorithms UEA2 & UIA2. Document 5: Design and Evaluation Report, Version 1.1, September 2006.

[10] O. Nyberg, J. Wall "Improved Linear Distinguishers for SNOW 2.0", *Proceedings of Fast Software Encryption,* LNCS 4047, pp. 144-162, 2006.

[11] D. Watanabe, A. Biryukov, C. De Canniere, "A Distinguishing Attack of SNOW 2.0 with Linear Masking Method", *Proceedings of Selected Areas in Cryptography*, LNCS 3006, pp. 222-233, 2004.

[12] B. Debraize, I.M. Corbella, "Fault Analysis of the Stream Cipher Snow 3G", *Proceedings of Workshop on Fault Diagnosis and Tolerance in Cryptography*, IEEE, pp. 103-110, 2009.

[13] B. Brumley, R. Hakala, K. Nyberg, B. Sovio, "Consecutive S-box Lookups: A Timing Attack on SNO 3G", *Proceedings of Information and Communications Security*, LNCS 6476, pp. 171-185, 2010.

[14] A. Biryukov, D. Priemuth-Schmid, B. Zhang, "Multiset collision attacks on reduced-round SNOW 3G and SNOW 3G", *Proceedings of Applied Cryptography and Network Security*, pp. 139-153, 2010.

[15] G. Orhanou, S. El Hajji, Y. Bentaleb,"NOW 3G stream cipher operation and complexity study", *Contemporary Engineering Sciences-Hikari Ltd*, 3(3), pp. 97-111, 2010.

[16] P. Kitsos, G. Selimis, O. Koufopavlou,"High performance ASIC implementation of the SNOW 3G stream cipher", *IFIP/IEEE VLSI-SOC,* 2008.

[17] O. Delgado-Mohatar, A. Fúster-Sabater,"Software Implementation of Linear Feedback Shift Registers over Extended Fields,' *Proceedings of CISIS/ICEUTE/SOCO Special Sessions,* pp. 117-126, 2012.

Criptosistemas de clave pública basados en acciones del anillo $E_p^{(m)}$

Joan-Josep Climent
Departament d'Estadística
i Investigació Operativa
Universitat d'Alacant
Email: jcliment@ua.es

Juan A. López-Ramos
Departamento de Matemáticas
Universidad de Almería
Email: jlopez@ual.es

Leandro Tortosa
Departament de Ciència de la
Computació i Intel·igència Artificial
Universitat d'Alacant
Email: tortosa@ua.es

***Abstract*—El objetivo de este trabajo es la introducción de aplicaciones criptográficas de una extensión del anillo $\mathrm{End}(\mathbb{Z}_p \times \mathbb{Z}_{p^2})$, denotado por $E_p^{(m)}$. Mostramos cómo las acciones del anillo $E_p^{(m)}$ sobre dos conjuntos distintos nos permiten introducir dos criptosistemas de clave pública diferentes y basados en la dificultad de resolver los problemas de la acción del semigrupo y de la descomposición respectivamente. Observamos cómo la no conmutatividad del anillo, así como la existencia de un gran número de divisores de cero lo hacen apropiado para tales aplicaciones criptográficas.**

***Palabras clave*—Criptosistema de Clave Pública (*Public Key Cryptosystem*), Problema de la Descomposición (*Decomposition Problem*), Problema de la Acción del Semigrupo (*Semigroup Action Problem*).**

I. Nomenclatura

SAP, Problema de la Acción del Semigrupo (Semigroup Action Problem).

DP, Problema de la Descomposición (Decomposition Problem).

$\mathrm{Mat}_{m\times m}(\mathbb{Z})$, matrices cuadradas de tamaño $m \times m$ sobre el anillo de los número enteros.

$r = (r_0, r_1, \ldots, r_{m-1})$, denota una matriz columna de m componentes.

E_p, anillo de matrices de tamaño 2×2 isomorfo al anillo $\mathrm{End}(\mathbb{Z}_p \times \mathbb{Z}_{p^2})$.

$E_p^{(m)}$, anillo de matrices de tamaño $m \times m$ y que es extensión del anillo E_p.

II. Introducción

Desde que en [4] Diffie y Hellman proponen el primer protocolo de intercambio de claves a través de un canal público, son muchos los autores que han introducido esquemas de este tipo, como por ejemplo [11], [12], [14], [16]. Si nos centramos en los criptosistemas de clave pública, podemos destacar indudablemente como los más conocidos [5] y [13]. Todos estos algoritmos, al igual que la idea original de Diffie y Hellman basan su fortaleza en la resolución de problemas sobre Teoría de Números. Por otro lado, trabajos como [15] o [9] en los que se muestran debilidades de [4], [5] y [13] y [12] respectivamente, así como los constantes avances en computación y capacidad de cálculo de las máquinas actuales han llevado a considerar otras estructuras para el desarrollo de tales protocolos de intercambio de clave y de criptosistemas de clave pública, tales como el conjunto de puntos de una curva elíptica ([7] y [10]) o cómo la acción de un semigrupo sobre un conjunto puede dar lugar también a este tipo de algoritmos [8]. En este caso, los autores muestran cómo es posible definir un criptosistema similar al de ElGamal [5] aprovechando la acción del semigrupo sobre un conjunto. Sin embargo, en este caso, un atacante debe resolver el conocido como *Problema de la Acción del Semigrupo* o SAP (*Semigroup Action Problem*), que consiste en dado un grupo abeliano finito G, un conjunto finito S y una acción de G sobre S, si dados $x, y \in S$ tales que $y = g \cdot x$ para algún $g \in G$, encontrar $h \in G$ tal que $y = h \cdot x$.

Basándose en el protocolo de intercambio de clave introducido en [16], Climent *et al.* en [2] introducen un protocolo de intercambio de clave para dos comunicantes sobre el anillo $\mathrm{End}(\mathbb{Z}_p \times \mathbb{Z}_{p^2})$. Sin embargo, aprovechando los elementos invertibles de anillo, Kamal y Youssef en [6] llevan a cabo un ataque a dicho protocolo de intercambio de claves. De este modo, y buscando evitar este tipo de ataques, en [1] los autores introducen un esquema general de intercambio de claves para un conjunto finito de usuarios, que extiende el de [2], pero que además se lleva a cabo sobre una extensión del anillo anterior, el anillo $E_p^{(m)}$, estudiado en [3], y en el que los elementos invertibles son muy escasos. La fortaleza de este nuevo intercambio de claves extiende el intercambio de clave de Diffie-Hellman, pero se basa en un problema mucho más duro que el del logaritmo discreto y que es conocido como *Problema de la Descomposición* o DP (*Decomposition Problem*), es decir, dado un grupo G, un subconjunto $S \subseteq G$ y un elemento $(x, y) \in G \times G$, encontrar elementos $z_1, z_2 \in S$ tales que $y = z_1 x z_2$.

El objetivo de este trabajo es usar la acción del semigrupo multiplicativo $E_p^{(m)}$ sobre un par de conjuntos para definir criptosistemas de clave pública en un entorno no conmutativo y basados en problemas distintos a los clásicos de Teoría de Números y de mucha más difícil resolución. En la sección III introducimos brevemente el anillo $E_p^{(m)}$, ampliamente estudiado en [3]. A continuación, en la sección IV y dado que los elementos de $E_p^{(m)}$ se expresan como matrices, usamos la acción de $E_p^{(m)}$ sobre una columna de elementos de un elemento en $E_p^{(m)}$ para definir un criptosistema basado en el

problema SAP. En la sección V, es la acción del propio anillo $E_p^{(m)}$ sobre sí mismo la que se usa para definir un criptosistema basado en el problema DP. La sección VI concluye los resultados introducidos en este trabajo.

III. El anillo $E_p^{(m)}$

En [2] los autores prueban que el anillo $\mathrm{End}(\mathbb{Z}_p \times \mathbb{Z}_{p^2})$ es isomorfo a un anillo que los autores denotan por E_p y cuyos elementos pueden representarse como matrices del tipo

$$\begin{bmatrix} a & b \\ pc & d \end{bmatrix} \quad \text{con } a, b, c \in \mathbb{Z}_p \text{ y } d \in \mathbb{Z}_{p^2}.$$

Motivados por los resultados citados en la introducción sobre el criptoanálisis llevado acabo sobre el anillo anterior, Climent *et al.* introducen en [3] la siguiente extensión del anillo E_p. Dado un primo p y un entero m, se define el conjunto

$$E_p^{(m)} = \Big\{ [a_{ij}] \in \mathrm{Mat}_{m\times m}(\mathbb{Z}) \mid a_{ij} \in \mathbb{Z}_{p^i} \text{ si } i \le j \text{ y } a_{ij} \in p^{i-j}\mathbb{Z}_{p^i} \text{ si } i > j \Big\}.$$

Asimismo se pueden definir una adición y una multiplicación como

$$[a_{ij}] + [b_{ij}] = [(a_{ij} + b_{ij}) \bmod p^i],$$
$$[a_{ij}] \cdot [b_{ij}] = \left[\left(\sum_{k=1}^{m} a_{ik} b_{kj} \right) \bmod p^i \right].$$

Se prueba entonces el siguiente resultado.

Teorema 1: *El conjunto $E_p^{(m)}$ con la adición y la multiplicación definidas anteriormente es un anillo no conmutativo.*

También en [3] se prueban las siguientes propiedades del anillo $E_p^{(m)}$ y que a continuación se relacionan.

Teorema 2: *El centro del anillo $E_p^{(m)}$ es el conjunto dado por*

$$Z(E_p^{(m)}) = \Big\{ [a_{ij}] \in E_p^{(m)} \mid a_{ij} = 0 \text{ si } i \neq j \text{ y } a_{ii} = \sum_{r=1}^{i} p^{i-r} u_{i-r} \text{ con } u_{i-r} \in \mathbb{Z}_p \Big\}.$$

Teorema 3: *Un elemento $[a_{ij}] \in E_p^{(m)}$ es invertible si y sólo si $a_{ii} \not\equiv_p 0$ para $i = 1, 2, \ldots, m$.*

Teorema 4: *El cardinal de $E_p^{(m)}$ es p^{ν_m} donde $\nu_m = (2m^3 + 3m^2 + m)/6$ y el cardinal del conjunto de los elementos invertibles viene dado por $p^{\nu_m - m}(p-1)^m$.*

Observación 1: A partir de los resultados anteriores y a modo de ejemplo, puede comprobarse que en $E_2^{(32)}$, el porcentaje de elementos no invertibles alcanza el 99.9999999767%.

IV. Un criptosistema de clave pública sobre $E_p^{(m)}$ basado en el SAP

El propósito de esta sección es la introducción de un criptosistema de clave pública basado en la acción del semigrupo multiplicativo de $E_p^{(m)}$ sobre el conjunto $\mathbb{Z}_p \times \mathbb{Z}_{p^2} \times \cdots \times \mathbb{Z}_{p^m}$. Dicha acción viene dada por la multiplicación definida sobre el anillo $E_p^{(m)}$, es decir, la acción que tiene lugar al multiplicar las filas del primer elemento por cada una de las columnas del segundo. Consideremos pues el anillo de polinomios $Z(E_p^{(m)})[X]$ con coeficientes sobre el centro del anillo $E_p^{(m)}$.

Algoritmo 1: *Sea $M \in E_p^{(m)}$ un valor público y $s \in \mathbb{Z}_p \times \mathbb{Z}_{p^2} \times \cdots \times \mathbb{Z}_{p^m}$ el mensaje que el comunicante Bernardo quiere enviar a Alberto de forma confidencial. Entonces:*

- *Alberto escoge $r \in \mathbb{Z}_p \times \mathbb{Z}_{p^2} \times \cdots \times \mathbb{Z}_{p^m}$, $f(X) \in Z(E_p^{(m)})[X]$ y calcula $t = f(M) \cdot r$.*
- *Alberto publica el par (r, t), manteniendo en secreto su propia clave privada $f(M)$.*
- *Bernardo elige $g(X) \in Z(E_p^{(m)})[X]$ y envía a Alberto el par*
$$(c_1, c_2) = (g(M) \cdot r, s + g(M) \cdot t).$$
- *Alberto obtiene $s = c_2 - f(M) \cdot c_1$.*

Notemos que aunque $E_p^{(m)}$ no es conmutativo, se tiene que $f(M)g(M) = g(M)f(M)$. De este modo se tiene el siguiente resultado.

Teorema 5: *El algoritmo 1 es correcto.*

Podemos observar que romper el algoritmo anterior involucra resolver el problema SAP, es decir, conocidos los valores r y $t = f(M) \cdot r \in \mathbb{Z}_p \times \mathbb{Z}_{p^2} \times \cdots \times \mathbb{Z}_{p^m}$, encontrar un elemento $A \in E_p^{(m)}$ que verifique $t = A \cdot r$. En [8] los autores prueban que cuanto más cerca se halle el semigrupo que actúa sobre el conjunto correspondiente de ser un grupo, es decir, cuanto más grande sea el conjunto de elementos invertibles en dicho semigrupo, más fácil es utilizar una versión sobre dicho semigupo del algoritmo del Polling-Hellman para el cálculo del logaritmo discreto. En nuestro caso, tal y como se ha indicado anteriormente, el número de elementos no invertibles puede acercarse mucho al total de elementos de $E_p^{(m)}$ con una adecuada elección de p y m.

Una característica adicional del criptosistema anterior es el uso de un polinomio distinto $g(X)$ cada vez que se cifra un mensaje. En el criptosistema de ElGamal, [5], la elección de un parámetro aleatorio para cada cifrado es necesaria para evitar un ataque texto claro-texto cifrado basado en la existencia de inversos. Sin embargo, en este caso, un ataque tal no podría llevarse a cabo debido a la ya citada escasa existencia de inversos. De hecho no puede llevarse a cabo ningún ataque basado en la división por no tratarse $E_p^{(m)}$ de un dominio de integridad. Además, la elección de dicho polinomio $g(X)$ de forma aleatoria evita ataques por repetición, pues un mismo mensaje no es cifrado dos veces del mismo modo.

Dada la clave pública (r, t), con $t = f(M) \cdot r$, sea A una solución del SAP, es decir, $A \cdot r = t$ y supongamos que el

elemento $g(M)$ verifica que $Ag(M) = g(M)A$. Entonces, dado el mensaje cifrado

$$(c_1, c_2) = (g(M) \cdot r, s + g(M) \cdot t)$$

tenemos que

$$\begin{aligned} & s + g(M) \cdot t - Ag(M) \cdot r \\ & = s + g(M)f(M) \cdot r - g(M)A \cdot r \\ & = s + g(M)f(M) \cdot r - g(M)f(M) \cdot r = s. \end{aligned}$$

De este modo es vital que el elemento $g(M)$ no conmute con la solución encontrada del SAP, A. Veamos que podemos tener condiciones para que ni $g(M)$ ni A sean elementos del centro de $E_p^{(m)}$.

La demostración del siguiente resultado es inmediata.

Lema 1: *Sea R un anillo no conmutativo. Entonces $Z(R[X]) = Z(R)[X]$.*

Una consecuencia directa del lema anterior nos da que $Z(E_p^{(m)}[X]) = Z(E_p^{(m)})[X]$.

Veamos ahora una condición para que $g(M)$ no sea un elemento central en $E_p^{(m)}$.

Lema 2: *Sea $g(X) = g_0 + g_1X + \cdots + g_kX^k \in Z(E_p^{(m)}[X])$ tal que $(g_j)_{i,i} \not\equiv_p 0$ para todo $i = 1, 2, \ldots, m$ y para algún $j = 1, 2, \ldots, k$. Si $M \notin Z(E_p^{(m)})$, entonces $g(M) \notin Z(E_p^{(m)})$.*

Demostración 1 (Demostración): Como $(g_j)_{i,i} \not\equiv_p 0$ para todo $i = 1, 2, \ldots, m$, tenemos que g_j es invertible. De este modo, si N es cualquier elemento en $E_p^{(m)}$ y suponemos que $Ng_jM = g_jMN$, entonces $g_jNM = g_jMN$, con lo que $MN = NM$, lo que es una contradicción. De este modo, aunque g_iM^i sea un elemento central para cualquier $i = 0, 1, 2, \ldots, k$ con $i \neq j$, tenemos que $g(M) \notin Z(E_p^{(m)})$.

Supongamos ahora que un atacante intenta encontrar A central y que sea solución del SAP $A \cdot r = t$. En tal caso, el elemento A ha de tener la forma

$$\operatorname{diag}(a_0, a_0 + pa_1, \ldots, a_0 + pa_1 + \cdots + p^{m-1}a_{m-1}),$$

con $a_i \in \mathbb{Z}_p$ para todo $i = 0, 1, 2, \ldots, m-1$.

Teorema 6: *Supongamos que*

$$\operatorname{diag}(a_0, a_0 + pa_1, \ldots, a_0 + pa_1 + \cdots + p^{m-1}a_{m-1}),$$

y que $r = (r_0, r_1, \ldots, r_{m-1})$ y $t = (t_0, t_1, \ldots, t_{m-1})$ son elementos de $\mathbb{Z}_p \times \mathbb{Z}_{p^2} \times \cdots \times \mathbb{Z}_{p^m}$ tales que $A \cdot r = t$. Si $r_i \not\equiv_p 0$ para $i = 0, 1, 2, \ldots, m-1$ entonces

$$r_0^{-1}t_0 \equiv_p r_1^{-1}t_1 \equiv_p \cdots \equiv_p r_{m-1}^{-1}t_{m-1}.$$

Demostración 2 (Demostración): De la igualdad $A \cdot r = t$ tenemos las siguientes igualdades

$$\begin{aligned} a_0r_0 &\equiv_p t_0, \\ (a_0 + pa_1)r_1 &\equiv_{p^2} t_1, \\ &\cdots \\ (a_0 + pa_1 + \cdots + p^{m-1}a_{m-1})r_{m-1} &\equiv_{p^m} t_{m-1}. \end{aligned}$$

Entonces $a_0 \equiv_p r_0^{-1}t_0$. De este modo $a_0 = r_0^{-1}t_0 + hp$ con $h \in \mathbb{Z}$. De la segunda expresión tenemos que $pa_1r_1 \equiv_{p^2} t_1 - a_0r_1$ y de este modo, $\operatorname{mcd}(pr_1, p^2) = p \mid (t_1 - a_0r_1)$. Por tanto, $t_1 \equiv_p a_0r_1$ y así $r_0^{-1}t_0 \equiv_p r_1^{-1}t_1$.

Supongamos ahora que

$$r_0^{-1}t_0 \equiv_p r_1^{-1}t_1 \equiv_p \cdots \equiv_p r_{m-2}^{-1}t_{m-2}.$$

De la expresión

$$a_0 + pa_1 + \cdots + p^{m-2}a_{m-2} \equiv_{p^{m-1}} r_{m-2}^{-1}t_{m-2},$$

tenemos que $a_0 + pa_1 + \cdots + p^{m-2}a_{m-2} = r_{m-2}^{-1}t_{m-2} + p^{m-1}h$ para algún $h \in \mathbb{Z}$. Entonces, como

$$(a_0 + pa_1 + \cdots + p^{m-1}a_{m-1})r_{m-1} \equiv_{p^m} t_{m-1}$$

tenemos que

$$a_0 + pa_1 + \cdots + p^{m-2}a_{m-2} + p^{m-1}a_{m-1} \equiv_{p^m} r_{m-1}^{-1}t_{m-1}$$

y por tanto $r_{m-2}^{-1}t_{m-2} + p^{m-1}(h + a_{m-1}) \equiv_{p^m} r_{m-1}^{-1}t_{m-1}$, de donde $r_{m-2}^{-1}t_{m-2} \equiv_p r_{m-1}^{-1}t_{m-1}$.

La consecuencia del resultado anterior es que dado el elemento $f(M)$, un usuario puede tomar como clave pública el par (r, t) con $r = (r_0, r_1, \ldots, r_{m-1})$ y $t = (t_0, t_1, \ldots, t_{m-1})$ con $r_i, t_i \in \mathbb{Z}_{p^{i+1}}$, para $i = 0, 1, 2, \ldots, m-1$ tales que $f(M) \cdot r = t$ y con r_k no divisible por p para ningún $k = 0, 1, 2, \ldots, m-1$ y de modo que exista i tal que $r_i^{-1}t_i \not\equiv_p r_0^{-1}t_0$, haciendo imposible que un atacante pueda calcular un elemento central A solución del SAP anterior.

Un ataque por fuerza bruta supondría probar con todos los posibles polinomios con coeficientes en el centro de $E_p^{(m)}$. Si consideramos un primo con 20 cifras decimales, o lo que es lo mismo, aproximadamente de 64 bits, y el entero $m = 5$, tenemos que el número de polinomios es del orden de 10^{121}. La longitud de los mensajes cifrados en este caso sería

$$2 \cdot (1024 + 512 + 256 + 128 + 64) = 9920 \text{ bits}.$$

V. Un criptosistema de clave pública sobre $E_p^{(m)}$ basado en el DP

La acción del semigrupo multiplicativo de $E_p^{(m)}$ sobre el propio conjunto $E_p^{(m)}$ nos define un criptosistema similar basado también en el SAP. Sin embargo, nuestro objetivo ahora es el de introducir un criptosistema de clave pública basado en los intercambios de clave introducidos en [1]. Consideramos la acción arriba indicada, aunque presentamos la siguiente variación. El algoritmo que introducimos, utilizando la misma notación que en el algoritmo 1, es como sigue:

Algoritmo 2: *Sea $M \in E_p^{(m)}$ un valor público y $S \in E_p^{(m)}$ el mensaje que el comunicante Bernardo quiere enviar a Alberto de forma confidencial. Entonces:*

- *Alberto escoge $N \in E_p^{(m)}$ tal que $MN \neq NM$, $f(X) \in Z(E_p^{(m)})[X]$ y un par de enteros positivos (u, v) y calcula $f(M)^uNf(M)^v$.*
- *Alberto publica el par $(N, f(M)^uNf(M)^v)$, manteniendo en secreto su propia clave privada $f(M)$ junto con los enteros u y v.*

- *Bernardo elige $g(X) \in Z(E_p^{(m)})[X]$ y un par de enteros positivos (r,t) y envía a Alberto el par*

$$(c_1,c_2) = \Big(g(M)^r N g(M)^t, \\ S + g(M)^r f(M)^u N f(M)^v g(M)^t\Big).$$

- *Alberto obtiene $S = c_2 - f(M)^u c_1 f(M)^v$.*

De nuevo, como en la sección anterior, el hecho de que $f(X)$ y $g(X)$ sean elementos del centro de $E_p^{(m)}[X]$ nos proporciona el siguiente resultado.

Teorema 7: *El algoritmo 2 es correcto.*

Teorema 8: *Romper el criptosistema dado por el algoritmo 2 es equivalente a resolver el problema DP.*

Demostración 3 (Demostración): Supongamos en primer lugar que somos capaces de resolver el problema DP dado por el par $(N, f(M)^u N f(M)^v)$. Entonces somos capaces de encontrar Z_1 y Z_2 tales que $g(M)^u N g(M)^v = Z_1 N Z_2$, $Z_1 M = M Z_1$ y $Z_2 M = M Z_2$. Entonces

$$\begin{aligned} c_2 - Z_1 c_1 Z_2 &= S + g(M)^r f(M)^u N f(M)^v g(M)^t \\ &\quad - Z_1 g(M)^r N g(M)^t Z_2 \\ &= S + g(M)^r f(M)^u N f(M)^v g(M)^t \\ &\quad - g(M)^r Z_1 N Z_2 g(M)^t \\ &= S + g(M)^r f(M)^u N f(M)^v g(M)^t \\ &\quad - f(M)^u g(M)^r N g(M)^t f(M)^v \\ &= S. \end{aligned}$$

Por otro lado, sean A y B elementos de $E_p^{(m)}$. Ciframos entonces el elemento B usando la clave pública $(A, f(M)^u A f(M)^v)$. El cifrado es entonces

$$(c_1,c_2) = \Big(g(M)^r A g(M)^t, \\ B + g(M)^r f(M)^u A f(M)^v g(M)^t\Big)$$

para $g(X)$, r y t como en el algoritmo 2.

Si recíprocamente estamos suponiendo que somos capaces de descifrar el mensaje anterior obteniendo B sin conocer la clave privada, entonces podemos calcular el elemento $c_2 - B$, lo que es equivalente a conocer la clave compartida por los comunicantes $g(M)^r f(M)^u A f(M)^v g(M)^t$, que es equivalente a resolver los problemas DP dados por los pares

$$(A, g(M)^r A g(M)^t) \quad \text{y} \quad (A, f(M)^u A f(M)^v).$$

Veamos ahora que el ataque introducido en [6] para romper el intercambio de claves dado en [2] no es posible en el caso del algoritmo 2, dependiendo de los parámetros escogidos. El ataque se basa en encontrar elementos W_1, W_2 de $E_p^{(m)}$ tales que

$$W_1 M = M W_1, \quad W_2 M = M W_2, \\ g(M)^r N g(M)^t W_2 = W_1 N.$$

Entonces tenemos que

$$\begin{aligned} W_1 f(M)^u N f(M)^v W_2^{-1} &= f(M)^u W_1 N W_2^{-1} f(M)^v \\ &= f(M)^u g(M)^r N g(M)^t f(M)^v \end{aligned}$$

puesto que $W_i f(M)^k = f(M)^k W_i$ para $i = 1, 2$ y cualquier valor de k. Por tanto, podemos calcular

$$c_2 - W_1 f(M)^u N f(M)^v W_2^{-1} = S.$$

Sin embargo, la existencia de W_2^{-1} en $E_p^{(m)}$ es casi improbable dependiendo de los valores de p y m tal y como ya se deja patente en la la observación 1.

VI. Conclusión

En este trabajo hemos mostrado cómo la acción de un semigrupo multicativo y no conmutativo puede ser usada para la definición de criptosistemas de clave pública cuya fortaleza reside en la resolución de problemas computacionalmente difíciles. En un caso obtenemos un criptosistema cuya seguridad se basa en el problema de la descomposición proveniente de la acción que define la multiplicación de elementos en el anillo no conmutativo $E_p^{(m)}$. Dado que los elementos de dicho anillo pueden representarse como matrices, si restringimos esta acción a las columnas de las mismas, obtenemos entonces un nuevo criptosistema basado esta vez en el problema de la acción del semigrupo. En ambos casos mostramos cómo las propias características del anillo en lo que se refiere a la no conmutatividad y a la gran existencia de elementos divisores de cero evitan diferentes posibles criptoanálisis.

Agradecimientos

El primer autor ha sido parcialmente financiado por el proyecto MTM2011-24858 del Ministerio de Economía y Competitividad del Gobierno de España. El segundo autor está financiado por el grupo de investigación de la Junta de Andalucía FQM 211.

Referencias

[1] J.-J. Climent, J.A. López-Ramos, P.R. Navarro, L. Tortosa, "Key agreement protocols for distributed secure multicast over the ring $E_p^{(m)}$," *WIT Transactions on Information and Communication Technologies*, vol. 45, pp. 13–24, 2013.

[2] J.-J. Climent, P.R. Navarro, L. Tortosa, "Key exchange protocols over noncommutative rings. The case of $\mathrm{End}(\mathbb{Z}_p \times \mathbb{Z}_{p^2})$," *International Journal of Computer Mathematics*, vol. 89(13-14), pp. 1753–1763, 2012.

[3] J.-J. Climent, P.R. Navarro, L. Tortosa, "An extension of the noncommutative Bergman's ring with a large number of noninvertible elements," enviado para su publicación.

[4] W.D. Diffie, M.E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22(6), pp. 644–654, 1976.

[5] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Transactions Information Theory*, vol. 31(4), pp. 469–472, 1985.

[6] A.A. Kamal, A.M. Youssef, "Cryptanalysis of a key exchange protocol based on the endomorphisms ring $\mathrm{End}(\mathbb{Z}_p \times \mathbb{Z}_{p^2})$," *Applicable Algebra in Engineering, Communications and Computing*, vol. 23(3-4), pp. 143–149, 2012.

[7] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48(177), pp. 203–209, 1987.

[8] M. Maze, C. Monico, J. Rosenthal, "Public Key Cryptography based on Semigroup Actions," *Advances in Mathematics of Communications*, vol. 1(4), pp. 489–507, 2007.

[9] A.J. Menezes, Y.H. Wu, "The discrete logarithm problem in $GL(n, q)$," *Ars Combinatoria*, vol. 47, pp. 23–32, 1997.
[10] V. Miller, "Use of Elliptic curves in Cryptography," *Advances in Cryptography – CRYPTO'85*, vol. 218, *Lecture Notes in Computer Science*, pp. 417–426. Springer-Verlag, New York, NY, 1986.
[11] A.G. Myasnikov, V. Shpilrain, A. Ushakov, "Group-based cryptography," Birkhäuser Verlag, 2008.
[12] R.W.K. Odoni, V. Varadharajan, P.W. Sanders, "Public key distribution in matrix rings," *Electronics Letters*, vol. 20, pp. 386–387, 1984.
[13] R.L. Rivest, A. Shamir, L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of ACM*, vol. 21(2), pp. 120–126, 1978.
[14] T. Satoh, K. Araki, "On construction of signature scheme over a certain non-commutative ring," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 80(1), pp. 40–45, 1997.
[15] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, 26(5), pp. 1484–1509, 1997.
[16] E. Stickel, "A new method for exchanging secret keys," *Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05),* Sidney, 2005, pp. 426-430.

Diseño de cifradores en flujo DLFSR con alta complejidad lineal para implementación hardware

A. Peinado
Universidad de Málaga
Andalucía Tech
ETSI Telecomunicación
Depto. Ingeniería de Comunicaciones
Email: apeinado@ic.uma.es

J. Munilla
Universidad de Málaga
Andalucía Tech
ETSI Telecomunicación
Depto. Ingeniería de Comunicaciones
Email: munilla@ic.uma.es

A. Fúster Sabater
Instituto de Tecnologías
Físicas y de la Información (ITEFI)
Consejo Superior de Investigaciones
Científicas (C.S.I.C.), Madrid
Email: amparo@iec.csic.es

***Resumen*—Muchos generadores de secuencias pseudoaleatorias de uso criptográfico se basan en registros de desplazamiento con realimentación dinámica (DLFSR) para incrementar el período y la complejidad lineal de las secuencias *PN*. En este trabajo se presenta un modelo teórico que permite el diseño de secuencias más largas y con mayor complejidad lineal que las obtenidas en otros esquemas de DLFSR. El modelo determina asimismo la relación constante entre período y complejidad lineal para estas estructuras. Las secuencias aquí obtenidas presentan mejores parámetros criptográficos que las de otras propuestas de registros de desplazamiento con realimentación dinámica encontradas en la literatura.**

***Palabras clave*—cifrado en flujo (*stream cipher*), complejidad lineal (*linear span*), generador de números pseudoaleatorios (*PRNG*), realimentación dinámica (*dynamic feedback*), registro de desplazamiento realimentado linealmente (*LFSR*), secuencia binaria (*binary sequence*).**

I. Introducción

Los registros de desplazamiento con realimentación lineal (LFSRs) se han utilizado tradicionalmente como bloques básicos para la implementación de generadores de secuencia con fines criptográficos [6]. Sus secuencias de salida, las *PN* secuencias, presentan buenas propiedades de pseudoaleatoriedad (equilibrio entre ceros y unos, excelente distribución de rachas, buena autocorrelación, etc.) pero son fácilmente previsibles debido a la linealidad inherente a estas estructuras. Con el fin de romper dicha linealidad, pero a la vez manteniendo las características de pseudoaleatoriedad, se aplican diferentes técnicas de diseño como son el filtrado no lineal, la decimación irregular de *PN* secuencias o la introducción de elementos típicos de los cifradores en bloque (cajas de sustitución, vueltas de generadores en bloque conocidos, funciones de expansión de claves, etc.).

Otra técnica general para romper la linealidad de los LFSRs consiste en la modificación dinámica de los parámetros de realimentación. Entre los diferentes ejemplos de aplicación de esta técnica pueden enumerarse los siguientes:

En 2008 Che *et al.* [3] propusieron una modificación del estado del LFSR para diseñar un generador de números aleatorios. Sin embargo, en 2011 este esquema fue rechazado cuando Meliá-Seguí *et al.* [11] detectaron ciertas debilidades que cuestionaban la aleatoriedad de la secuencia de salida. A su vez, Hellebrad [8] y Rosinger [16] propusieron sendos generadores de secuencia para testeo de circuitos basados en modificaciones dinámicas de las semillas (estados iniciales) y de los polinomios de realimentación de los LFSRs.

En 2002 Mita *et al.* [12] diseñaron un generador de secuencia pseudoaleatoria basado en un LFSR con realimentación dinámica cuyo polinomio de realimentación se actualizaba según fuera el estado de otro LFSR secundario. Esta estructura puede considerarse como el inicio de los generadores DLFSR (Dynamical LFSR). Posteriormente en 2005, Babbage *et al.* diseñaron el cifrador en flujo Mickey [1] compuesto por dos LFSRs conectados entre sí de manera que cada uno de ellos controlaba la realimentación del otro. Sin embargo en 2003 Ding *et al.* [5] criptoanalizaron dicho generador.

En 2007 Kiyomoto *et al.* [9] propusieron el cifrador K2, basado en dos LFSRs y un filtro no lineal. En dicho cifrador, un bit del estado del LFSR secundario controlaba la realimentación del LFSR principal. Posteriormente, Bogdanov *et al.* [2] presentaron una evaluación positiva de la seguridad del cifrador K2.

El generador Rakaposhi fue propuesto en 2009 por Cid *et al.* [4]. Se compone de un LFSR cuyo polinomio de realimentación se selecciona entre 4 posibles opciones codificadas por 2 bits del estado de un registro con realimentación no lineal (NLFSR). La secuencia de salida se obtiene aplicando un filtro no lineal a ambos registros (LFSR y NLFSR). Recientemente, en 2013 Orumiehchiha *et al.* [13] han detectado ciertas vulnerabilidades en dicho cifrador.

En 2013, se presentó un generador de números aleatorios, el J3Gen [10], que utilizaba un LFSR cuyo polinomio de realimentación se seleccionaba de una lista de polinomios mediante un esquema cíclico. También en 2013 Peinado *et al.* [14] desarrollaron un modelo matemático, basado en secuencias entrelazadas [7], para calcular el período y la complejidad lineal de los generadores DLFSR. Posteriormente, dicho modelo se aplicó al generador pseudoaleatorio descrito en [12].

En este trabajo, se presenta una extensión del modelo matemático para DLFSRs que permite generar secuencias con mayor período y complejidad lineal que aquellas obtenidas en las referencias anteriores. Los resultados aquí descritos

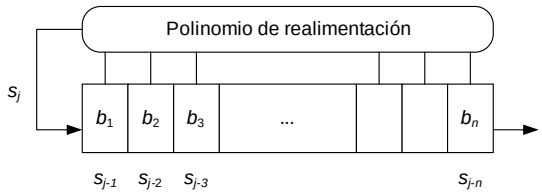


Figura 1. Estructura de un LFSR

mejoran la seguridad y robustez de las propuestas basadas en módulos DLFSR.

II. Registros de desplazamiento con realimentación dinámica (DLFSR)

Un LFSR es un registro de desplazamiento compuesto por n celdas de memoria con contenido binario $b_1, b_2, \ldots, b_n$ que funcionan de forma síncrona. A cada golpe de reloj, el contenido de cada celda se desplaza una posición a la derecha según se observa en Fig. 1. Mediante una función de realimentación representada por el polinomio de realimentación se genera un nuevo contenido para la celda b_1. En este trabajo se considerarán únicamente celdas con contenido binario, sin embargo también se han diseñado registros de desplazamiento cuyas celdas contienen elementos en un cuerpo de Galois extendido $GF(2^n)$. Si el bit de salida de la función de realimentación en el instante j es s_j, entonces el estado del LFSR de n celdas que produce s_j es $(s_{j-1}, s_{j-2}, \ldots, s_{j-n})$. Por tanto,

$$s_j = c_1 s_{j-1} + c_2 s_{j-2}, \ldots, c_n s_{j-n}, \qquad (1)$$

donde $c_1, \ldots, c_n$ son los coeficientes binarios del polinomio de realimentación

$$p(x) = c_n x^n + c_{n-1} x^{n-1} + \ldots + c_2 x^2 + c_1 x + 1. \qquad (2)$$

Es bien conocido [6] que el máximo período de la secuencia de salida de un LFSR es $2^n - 1$, es decir el registro pasa por todos los posibles estados distintos de cero. Esto sucede cuando el polinomio de realimentación es un polinomio primitivo, en cuyo caso la secuencia generada por el LFSR es una *PN* secuencia o secuencia de longitud máxima. Dichas secuencias presentan un perfecto equilibrio y distribución estadística de ceros y unos a la vez que una autocorrelación bivaluada. Es decir satisfacen perfectamente los postulados de pseudoaleatoriedad de Golomb [6]. Sin embargo el conocimiento de solamente $2n$ bits de dicha secuencia permite reconstruirla en su totalidad, ya que los coeficientes del polinomio de realimentación pueden obtenerse como solución de un sistema de n ecuaciones lineales.

Un DLFSR es un tipo de LFSR en el que el polinomio de realimentación va cambiando a medida que el registro se va desplazando. Tal y como se muestra en la Fig. 2, el modelo conceptual de un DLFSR consiste en un LFSR principal más un módulo adicional que controla el instante en el que se aplica

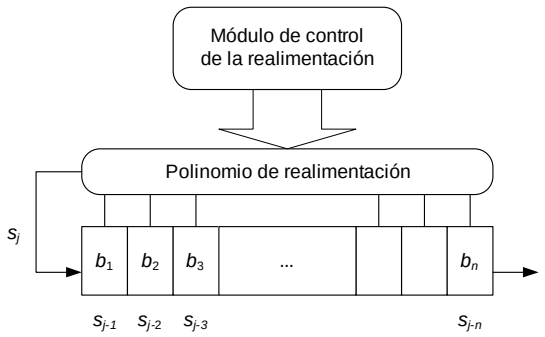


Figura 2. Estructura de un DLFSR

un nuevo polinomio de realimentación. Las secuencias generadas por un DLFSR pueden considerarse como la concatenación de segmentos de *PN* secuencias, de tal forma que el estado final del LFSR correspondiente al polinomio de realimentación $p_i(x)$ coincide con el estado inicial del LFSR correspondiente al polinomio de realimentación $p_{i+1}(x)$.

La finalidad de un DLFSR es la de generar secuencias con un período más largo y una complejidad lineal mayor que las producidas por el simple LFSR. Para llevar a cabo esta tarea, el modulo de control modifica diferentes parámetros de realimentación generando así una secuencia distinta. Los parámetros de realimentación de un DLFSR pueden enumerarse tal y como sigue:

- n : Longitud del LFSR a la vez que el grado del polinomio de realimentación.
- N_p : Número de diferentes polinomios de realimentación que van a aplicarse sobre el LFSR. En general estos polinomios son primitivos aunque también existen esquemas que incluyen polinomios no primitivos. La mayoría de diseños de DLFSR seleccionan polinomios con muchos coeficientes comunes para simplificar así la implementación hardware.
- e_i : Número de bits consecutivos generados por el mismo polinomio $p_i(x)$. Este parámetro puede ser fijo o variable.
- *Regla de selección*: Determina el orden en el que se aplican los distintos polinomios de realimentación. En algunos casos se aplican siguiendo un orden predeterminado; en otros, de forma completamente aleatoria.
- *Módulo de control*: El módulo de control establece el modo en el que se establece el instante en el que se cambia el polinomio de realimentación, así como el polinomio a utilizar o el número de bits que se van a generar. Este modo puede ser dependiente o independiente del LFSR principal, empleando señales externas al LFSR principal, como por ejemplo, LSFR secundarios; o utilizando el contenido de determinadas celdas del propio LFSR principal. Los mayores valores de complejidad lineal se alcanzan cuando se utilizan dispositivos adicionales, y por tanto, esquemas de control independientes.

Como ejemplo ilustrativo analizamos el DLFSR propuesto por Mita *et al.* en [12] que consta de un LFSR principal con $n = 16$ celdas y $N_p = 4$ polinomios de grado 16. La

regla de selección utilizada establece que los N_p polinomios se aplican siempre en el mismo orden, generanado cada uno de ellos siempre el mismo número de bits. El modulo de control está compuesto por un LFSR secundario de $m = 5$ celdas y polinomio de realimentación primitivo de grado 5. El LFSR secundario se conecta a un decodificador que, de acuerdo con su estado actual y una regla de selección fija, toma de una tabla el correspondiente polinomio a aplicar sobre el LFSR principal. A cada polinomio se le asigna un único estado del LFSR secundario. Cuando este LFSR alcanza dicho estado, el polinomio $p_i(x)$ actua sobre el LFSR principal. Por tanto sólo 4 estados del secundario modifican la realimentación del registro principal. Los 4 polinomios se aplican siguiendo una rotación cíclica. Sumando el número de bits consecutivos e_1, e_2, e_3, e_4 generados respectivamente por $p_1(x), p_2(x), p_3(x), p_4(x)$, se obtiene el período completo de la secuencia producido por el LFSR secundario, esto es

$$\sum_{i=1}^{4} e_i = 2^5 - 1 = 31. \tag{3}$$

Por otra parte, el cifrador en flujo Rakaposhi [13] es un DLFSR compuesto por un LFSR principal de longitud $n = 192$ bits con $N_p = 4$ polinomios de realimentación de grado 192. El módulo de control es un registro de desplazamiento con realimentación no lineal (NLFSR) de 128 celdas, dos de las cuales se emplean para seleccionar el polinomio de realimentación del LFSR principal. Por tanto, e_1, e_2, e_3, e_4 se determinan dinámicamente mediante los valores que toman 2 bits del NLFSR.

III. Período y complejidad lineal de las secuencias generadas

Desde un punto de vista criptográfico, el período y la complejidad lineal son dos indicadores fundamentales de la pseudoaleatoriedad de una secuencia. Ambas propiedades se definen tal y como sigue.

Definición 1. Sea $s = (s_0, s_1, s_2, \ldots) = (s(t))\ t \geq 0$ una secuencia binaria. Si existe un entero $r > 0$ tal que $s(t) = s(t + r)$ para todo $t \geq 0$, entonces la secuencia s se dice periódica y su período, representado por $T(s)$, es r.

Definición 2. La complejidad lineal de una secuencia s, representada por LC, es la longitud del LFSR más corto que puede generar dicha secuencia.

Para determinar el período y la complejidad lineal de las secuencias producidas por DLFSRs, hay que tener en cuenta que dichas secuencias son *secuencias entrelazadas* en el sentido dado en [7]. Es decir que la secuencia de salida de un DLFSR puede descomponerse en diferentes secuencias decimadas, todas ellas generadas por el mismo polinomio de realimentación. Una secuencia decimada $w_j(t)$ se construye tomando uno de cada N_s bits de la secuencia $s(t)$ empezando en $s(j)$, es decir $w_j(t) = s(j + tN_s)\ t \geq 0$. Este hecho ya fue señalado en [14] dando lugar a un modelo matemático para la generación de números pseudoaleatorios mediante DLFSRs, que puede resumirse en la siguiente ecuación:

$$M = \prod_{t}^{t+N_s} A_t, \tag{4}$$

donde A_t es una matriz $n \times n$ cuyo polinomio característico es el polinomio $p_t(x)$ aplicado al DLFSR en el instante t. El parámetro N_s es el período de aplicación de los distintos polinomios de realimentación a la vez que coincide con el número de secuencias decimadas que constituyen la secuencia entrelazada. Como ejemplo ilustrativo, podemos decir que el DLFSR definido en [12] utiliza 4 polinomios $p_1(x)$, $p_2(x)$, $p_3(x)$ y $p_4(x)$ de la siguiente manera: $p_1(x)$ genera 9 bits consecutivos, $p_2(x)$ 5 bits, $p_3(x)$ un único bit y $p_4(x)$ genera 16 bits consecutivos de su correspondiente *PN* secuencia. A continuación $p_1(x)$ generaría de nuevo 9 bits y así sucesivamente. Por tanto en este ejemplo tendríamos $N_s = 31$ y la ecuación (4) se podría reescribir como

$$M = \prod_{i=1}^{4} A_{p_1}^{9} \cdot A_{p_2}^{5} \cdot A_{p_3} \cdot A_{p_4}^{16}, \tag{5}$$

donde A_{p_i} es una matriz $n \times n$ cuyo polinomio característico es el polinomio $p_i(x)$ aplicado al DLFSR.

El polinomio característico $c_M(x)$ de la matriz M determina el período T_M de las secuencias decimadas. Nótese que las N_s secuencias decimadas tienen el mismo polinomio característico [14]. Por tanto, el período T de la secuencia total viene dado por

$$T = T_M \cdot N_s. \tag{6}$$

Por otro lado, el DLFSR establece que las secuencias decimadas se generan con un LFSR de n etapas, luego la complejidad lineal de estas secuencias es n y la complejidad lineal total es

$$LC = n \cdot N_s. \tag{7}$$

IV. Generación de secuencias con mayores períodos y complejidades

A partir de las ecuaciones (6) y (7) se observa que si el polinomio característico $c_M(x)$ fuera primitivo, entonces se alcanzaría el período máximo. En algunos casos, por ejemplo en [12], se puede seleccionar el binomio (N_p, e_i) óptimo para obtener un polinomio $c_M(x)$ primitivo [14]. Sin embargo en otros casos, la única manera de aumentar el período consiste en incrementar el número N_s de secuencias decimadas. Al mismo tiempo, el incremento de N_s también aumenta el valor de la complejidad lineal. Nótese sin embargo que la relación complejidad lineal/período o n/T_M es constante para todo esquema DLFSR; es decir la razón n/T_M es la misma en las secuencias generadas por un DLFSR o por un simple LFSR.

Aunque todos los generadores basados en DLFSR persiguen aumentar el número de secuencias decimadas N_s, se pueden agrupar en distintas categorías en función del modo en que se consigue dicho aumento. Así, tanto en el sistema presentado en [12], como en la configuración del sistema propuesto en [10] cuando $l = 1$, se utiliza un número de polinomios N_p que se aplican siempre en el mismo orden, generando cada

uno de ellos un número de bits constante y determinando un modo de funcionamiento definido por la siguiente expresión

$$M = \prod_{i=1}^{N_p} A_{p_i}^{e_i}, \tag{8}$$

lo que determina que el número total de secuencias decimadas N_s sea

$$N_s = \sum_{i=1}^{N_p} e_i, \tag{9}$$

En otras ocasiones, como en [15], o en el caso general ($l > 1$) de [10], los sistemas DLFSR aumentan N_s utilizando los distintos polinomios siempre en el mismo orden, pero generando cada uno de ellos un número de bits diferente cada vez que se aplican. La expresión que describe su funcionamiento es la siguiente:

$$M = \prod_{i=1}^{L_m} A_{p_{i \bmod N_p}}^{e_i}, \tag{10}$$

siendo L_m la longitud de una secuencia pseudoaleatoria secundaria, lo que determina un número de secuencias decimadas $N_s = \text{mcm}(N_r, N_p)$, donde

$$N_r = \sum_{i=1}^{L_m} e_i, \tag{11}$$

donde $e_i < n$. Por último, propuestas como [1], [4], [9], utilizan un conjunto de polinomios que se aplican siguiendo un orden pseudoaleatorio, pero generando un único bit en cada ocasión. El modelo que define el funcionamiento viene determinado por la siguiente expresión

$$M = \prod_{i=1}^{N_s} A_{p_{z_i}}, \tag{12}$$

donde $1 \leq z_i \leq N_p$, siendo z_i el valor determinado en el instante i por una secuencia pseudoaleatoria de longitud N_s generada por el modulo de control de realimentación del DLFSR. En consecuencia, la matriz M es difícilmente calculable, aunque el período y la complejidad lineal se pueden estimar con las mismas expresiones generales de las categorías anteriores.

De las tres categorías reseñadas, la primera ecuación (8) genera secuencias con una estructura interna que presenta una alta linealidad, lo que facilita su criptoanálisis. La segunda ecuación (10) es la que permite generar períodos mayores. La tercera categoría ecuación (12), aunque presenta períodos menores que las anteriores, tiene una estructura interna mucho más robusta.

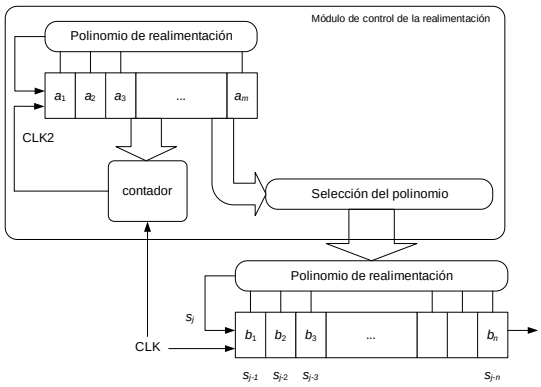


Figura 3. Arquitectura de DLFSR con esquema dinámico variable

IV-A. Generador propuesto basado en DLFSR

En esta sección se propone un diseño de DLFSR genérico, en el que se combinan aspectos de la segunda y tercera categoría, manteniendo unos valores elevados de complejidad lineal y período con una estructura interna más robusta. Utilizando la notación introducida en la sección II, este sistema utiliza un LFSR de n celdas y N_p polinomios, donde cada polinomio $p_i(x)$ genera un número pseudoaleatorio de bits cada vez que se aplica. La regla de selección establece que el orden de aplicación de los polinomios sea pseudoaleatorio, y el módulo de control se compone de un LFSR adicional y un contador, lo que determina un esquema independiente del LFSR principal (Fig. 3). A continuación de describen los componentes del sistema propuesto.

LFSR principal. Es un LFSR de n etapas con N_p polinomios de realimentación primitivos que se irán aplicando siguiendo un orden pseudoaleatorio marcado por el LFSR secundario.

LFSR secundario. Es un LFSR controlado por reloj, de longitud m y polinomio primitivo de grado m, que controla el polinomio de realimentación del LFSR principal mediante a) k_1 de sus bits para la selección de dicho polinomio y b) un contador decreciente que se inicializa con $k_2 \leq \log_2 n$ de sus bits.

Contador. Es un contador que realiza siempre una cuenta atrás completa a partir del valor que determina el LFSR secundario. Al mismo tiempo, el contador controla el reloj de este LFSR secundario. De este modo, cada vez que el contador llega a cero, el LFSR secundario genera un nuevo bit, cambia su estado y reinicializa el contador con un nuevo valor. En ese mismo instante, el polinomio de realimentación del LFSR principal se modifica, puesto que está controlado por k_2 bits del LFSR secundario.

El proceso de generación se detalla tal y como sigue:

- Los LFSRs se inicializan con las correspondientes semillas.
- El contador se inicializa con el estado de k_2 bits del LFSR secundario.
- El LFSR principal comienza a generar bits con el polinomio que determinan k_1 bits del LFSR secundario.

- Simultáneamente el contador comienza la cuenta atrás hasta alcanzar el valor 0. En ese momento, se activa la señal de reloj CLK2 para que el LFSR secundario genere un nuevo bit.
- El nuevo estado del LFSR secundario determina, mediante k_1 bits el nuevo polinomio del LFSR principal, y mediante k_2 el nuevo valor del contador para que inicie la cuenta atrás.

En consecuencia, tanto el orden en el que se aplican los polinomios de realimentación como los bits que genera cada uno vendrán determinados mediante una secuencia pseudo-aleatoria generada por el LFSR secundario. De esta forma, el comportamiento del DLFSR queda determinado por la expresión

$$M = \prod_{i=1}^{L_m} A_{p_{z_i}}^{e_i}, \tag{13}$$

siendo $L_m = 2^m - 1$ la longitud de la secuencia generada por el LFSR secundario, y el número de secuencias decimadas $N_s = N_r$ se calcula aplicando (11). Además, $1 \leq z_i \leq N_p$, siendo z_i el valor determinado en el instante i por k_1 celdas del LFSR secundario.

V. Comparación de resultados

La mejora de la complejidad lineal y del período de las secuencias generadas por el DLFSR propuesto en este trabajo queda patente a través de las expresiones de la sección anterior. Con el fin de ilustrar esta mejora se han realizado diversas comparaciones con algunos de los DLFSR citados previamente. Dado que la relación entre la complejidad lineal y el período de las secuencias generadas por un DLFSR se mantiene constante, las comparaciones se han realizado sobre el número N_s de secuencias decimadas, que es el valor que determina el incremento tanto de la complejidad lineal como del período. Así, el DLFSR(15,6) propuesto por Mita *et al* en [12], perteneciente a la primera categoría de DLFSRs (8), compuesto de un LFSR principal de 16 celdas, otro secundario de 5 celdas y cuatro polinomios de realimentación, presenta un valor $N_s = 2^5 - 1 = 31$ determinado por la longitud de la secuencia generada por el LFSR secundario.

Si configuramos el generador propuesto en este trabajo con los mismos valores, LFSR principal de 16 celdas y secundario de 5 celdas, se tendrían que utilizar $k_1 = 2$ celdas del LFSR secundario para seleccionar los $N_p = 4$ polinomios de realimentación, y $k_2 = 4$ celdas para indicar el número de bits que generará cada polinomio. Es importante recordar que cada polinomio de realimentación no debe generar más de n bits consecutivos, siendo n el grado del polinomio, para evitar posibles criptoanálisis. El número de secuencias decimadas N_s, según indica la ecuación (11) es la suma N_r del valor decimal de todos los estados por los que pasan los k_2 bits. En este caso, $k_2 = 4$ determina la selección de 4 celdas de entre las 5 del LFSR secundario. Por tanto, el período de estas $k_2 = 4$ celdas seguirá siendo $2^5 - 1$, aunque la suma de los estados que se suceden se aproxima por la siguiente expresión

Tabla I
Comparación de resultados

Generador	N_p	N_s
Rakaposhi (192,128)	4	2^{128}
Nuevo DLFSR (192,128)	4	$(2^{128} - 1) \cdot (2^6)$
Mita DLFSR (192,128)	4	$2^{128} - 1$

$$N_s = N_r \leq (2^5 - 1) \cdot (2^{k_2} - 1) = 248, \tag{14}$$

lo que supone una mejora de aproximadamente un orden de magnitud. Sin embargo, estas configuraciones no se corresponden con los valores de las implementaciones reales. Por ello, se ha realizado una comparación (tabla I) con el DLFSR utilizado en el cifrador Rakaposhi [4] que utiliza un LFSR principal de 192 celdas y un NLFSR secundario de 128. Al pertenecer este DLFSR a la segunda categoría (10), el número de secuencias decimadas N_s se corresponde directamente con la longitud de la secuencia generada por el NLFSR, es decir, $N_s = 2^{128}$. Si se configura el DLFSR propuesto con un LFSR principal de 192 celdas y un LFSR secundario de 128, se necesitarían $k_1 = 2$ celdas para seleccionar los $N_p = 4$ polinomios y $k_2 = 7$ celdas para indicar el número de bits consecutivos que cada polinomio debe generar. En consecuencia, como indica la tabla I, el número de secuencias N_s se puede aproximar como

$$N_s = N_r \leq (2^{128} - 1) \cdot (2^{k_2 - 1}) \tag{15}$$

El DLSFR propuesto en este trabajo incrementa N_s, y por tanto la complejidad lineal y el período, en un factor de $(2^{k_2} - 1)$ con respecto a la que se obtiene con el cifrador Rakaposhi. Por último, para completar la comparación, se analiza el DLFSR genérico propuesto en [12] para los mismos valores del cifrador Rakaposhi. Se obtiene $N_s = 2^{128} - 1$, que es muy similar al valor obtenido para el cifrador Rakaposhi. La diferencia reside en la estructura interna de las secuencias generadas en [12], que permite a un atacante criptoanalizarlas con facilidad.

VI. Conclusiones

En la actualidad numerosos generadores de secuencia cifrante para uso criptográfico pertenecen al grupo de generadores DLFSR.

En este trabajo se ha desarrollado un nuevo tipo de generador DLFSR que mejora la complejidad lineal y el período de las secuencias producidas. Para ello, se ha partido de una clasificación de los sistemas basados en DLFSR en función de la técnica empleada para aumentar N_s (el número de secuencias decimadas que conforman la secuencia entrelazada global); se ha modelado su funcionamiento a partir del modelo propuesto en [14]; y se ha propuesto un nuevo tipo que combina las ventajas de los anteriores, mejorando complejidad lineal y período y disminuyendo la linealidad de la estructura interna del generador.

El esquema propuesto está formado por dos LFSRs y un contador combinados de manera que el reloj del LFSR principal está controlado por el contador, que a su vez está controlado por el LFSR secundario. El efecto que se consigue es que los polinomios de realimentación se apliquen siguiendo un orden pseudoaleatorio y que cada uno de estos polinomios genere un número de bits consecutivos determinados también de forma pseudoaleatoria.

Por último se han comparado los valores del parámetro N_s que utilizan algunos de los generadores basados en DLFSR que se han propuesto recientemente, como es el caso del cifrador Rakaposhi.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el MICINN en el marco del proyecto "TUERI: Tecnologías segUras y Eficientes para las Redes inalámbricas en la Internet de las cosas con aplicaciones en transporte y logística", TIN2011-25452; y por la Universidad de Málaga, Andalucía Tech.

REFERENCIAS

[1] S. Babbage, M. Dodd, "The MICKEY Stream Ciphers," in *New Stream Cipher Designs. The eSTREAM Finalists*, M. Robshaw, O. Billet, Eds. LNCS 4986, Springer-Verlag, 2008, pp. 191–209.

[2] A. Bogdanov, B. Preneel, V. Rijmen, "Security Evaluation of the K2 Stream Cipher," Internal report, Katholieke Universiteit Leuven, ESAT/SCD-COSIC, March 2011.

[3] W. Che, H. Deng, X. Tan, J. Wang, "A Random Number Generator for Application in RFID Tags," in *Networked RFID Systems and Lightweight Cryptography*. Springer: Berlin/Heidelberg, Germany, 2008, Chapter 16, pp. 279–287.

[4] C. Cid, S. Kiyomoto, J. Kurihara, "The RAKAPOSHI Stream Cipher," in *Information and Communications Security*, LNCS 5927[C], Springer-Verlag, 2009, pp. 32–46.

[5] L. Ding, J. Guan, "Cryptanalysis of Mickey family of stream ciphers," *Security and Communication Networks*, vol. 6 (8), pp. 936–941, 2013.

[6] S.W. Golomb, "Shift-Register Sequences," revised edition, Aegean Park Press, Laguna Hill, California, 1982.

[7] G. Gong, "Theory and Applications of q-ary interleaved sequences," *IEEE Transactions on Information Theory*, vol. 41 (2), pp. 400–411, 1995.

[8] S. Hellebrand, J. Rajskia, S. Tarnick, S. Venkataraman, B. Courtois, "Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers," *IEEE Trans. Comput.*, vol. 44, pp. 223–233, 1995.

[9] S. Kiyomoto, T. Tanaka, K. Sakurai, "K2: A stream cipher algorithm using dynamic feedback control," in *Proceedings of SECRYPT*, J. Hernando, E. Fernández-Medina, M. Malek, Eds. INSTICC Press, 2007, pp. 204–213.

[10] J. Meliá-Seguí, J. García-Alfaro, J. Herrera-Joancomartí, "J3Gen: A PRNG for Low-Cost Passive RFID," *Sensors*, vol. 13, pp. 3816– 3830, 2013.

[11] J. Meliá-Seguí, J. García-Alfaro, J. Herrera-Joancomartí, "A practical implementation attack on weak pseudorandom number generator designs for EPC Gen2 Tags," *Wirel. Pers. Commun.*, vol. 59, pp. 27– 42, 2011.

[12] R. Mita, G. Palumbo, S. Pennisi, M. Poli, "Pseudorandom bit generator based on dynamic linear feedback topology," *Electronic Letters*, vol. 38 (19), pp. 1097– 1098, 2002.

[13] M.A. Orumiehchiha, J. Pieprzyk, E. Shakour, R. Steinfeld, "Security Evaluation of Rakaposhi Stream Cipher," in *Information Security Practice and Experience*, 9th International Conference, ISPEC 2013, R. Deng, T. Feng, Eds. LNCS 7863, Springer-Verlag, 2013, pp. 361–371.

[14] A. Peinado, A. Fúster-Sabater, "Generation of pseudorandom binary sequences by means of LFSRs with dynamic feedback," *Mathematical and Computer Modelling*, vol. 57 (11-12), pp. 2596–2604, 2013.

[15] A. Peinado, J. Munilla, A. Fúster-Sabater, "Improving the Period and Linear Span of the Sequences Generated by DLFSRs," *7th International Conference on Computational Intelligence in Security for Information Systems, CISIS 2014*, Bilbao, Spain, 25th-27th June, 2014

[16] P. Rosinger, B. Al-Hashimi, N. Nicolici, "Dual multiple-polynomial LFSR for low-power mixed-mode BIST," *IEEE Proc. Comput. Digital Tech.*, vol. 150 , pp. 209–217, 2003.

Privacy-Preserving Group Discounts

Josep Domingo-Ferrer
Departament d'Enginyeria
Informàtica i Matemàtiques
Universitat Rovira i Virgili
Email: josep.domingo@urv.cat

Alberto Blanco-Justicia
Departament d'Enginyeria
Informàtica i Matemàtiques
Universitat Rovira i Virgili
Email: alberto.blanco@urv.cat

***Abstract*—How can a buyer legitimately benefit from group discounts while preserving his privacy? We show how this can be achieved when buyers can use their own computing device (*e.g.* smartphone or computer) to perform a purchase. Specifically, we present a protocol for privacy-preserving group discounts. The protocol allows a group of buyers to prove how many they are without disclosing their identities. Coupled with an anonymous payment system, this allows group discounts to be compatible with buyer privacy.**

***Keywords*—Buyer privacy, Group discounts, Cryptographic protocols, Digital signatures**

I. Introduction

Group discounts are offered by vendors to encourage consumers to use their services, to promote more efficient use of resources, to protect the environment, etc. Examples include group tickets for museums, stadiums or leisure parks, discounted highway tolls or parking fees for high-occupancy vehicles, etc. It is common for the vendor to require all group members to identify themselves, but in reality this is seldom strictly necessary.

We make the assumption that the important feature about the group is the *number of its members*, rather than their identities. A secondary feature that may often (not always) be relevant for a group discount is whether group members are physically together.

Anonymously proving the number of group members and their being together is trivial in a face-to-face setting with a human verifier, who can see that the required number of people are present. However, with an automatic verifier and/or in an on-line setting, this becomes far from obvious.

In this paper, we propose a method to prove the number of people in a group while preserving the anonymity of group members and without requiring specific dedicated hardware, except for a computing device with some wireless communication capabilities (*e.g.* NFC, Bluetooth or WiFi). Also, we explore the option to include payment in our proposed system, which is necessary for group discounts. We complete the description of our method with a possible anonymous payment mechanism, *scratch cards*. The method presented here is a generalization of a specific protocol for toll discounts in high-occupancy vehicles, whose patent we recently filed [6].

The rest of the paper is structured as follows. Section II describes the building blocks of our method, namely a digital signature scheme, a key management scheme, an anonymous payment scheme and wireless communication technologies; the latter technologies should be short-range in applications where one wants to check that the group members are physically together. Section III describes our actual group size accreditation method, including the required entities and protocols. In Section IV, we give a complexity estimation of our proposal. Section V sketches conclusions and future work ideas.

II. Building Blocks

Our group size accreditation method is based on an identity-based dynamic threshold (*IBDT*) signature scheme, namely a particular case of the second protocol proposed in [7].

Threshold signature schemes are commonly based on (t, n)-threshold secret sharing schemes, such as the ones introduced in [1] and [12], and they require a minimum number t of participants to produce a valid signature. Dynamic threshold signature schemes differ from the previous ones in that the threshold t is not fixed during the setup phase, but is declared at the moment of signing. Our method takes advantage of this feature to find out how many users participated in the signature of a particular message, and consequently how many people form a group. If one wishes to prove that the signature is not only computed by at least t participants, but also that *these are together in the same place*, the above signature schemes need to be complemented with short-range communication technologies.

On the other hand, identity-based public key signature schemes, theorized by Shamir in [13] and with the first concrete protocol, based on the Weil pairing, developed by Boneh *et al.* in [3], allow public keys pk^U to be arbitrary strings of some length, which we call *identities*. These strings are associated with a user U and reflect some aspect of his identity, *e.g.* his email address. The corresponding secret key sk^U is then computed by a trusted entity, the certification authority (CA), taking as input the user's identity and, possibly, some secret information held only by the CA, and is sent to the user U through some secure channel. Identity-based public key signature schemes offer a great flexibility in key generation and management and our method takes advantage of this feature by proposing a key management scheme that allows preserving the anonymity of the participants.

Finally, in most group discounts, a fee must be paid after proving the number of group members, so an anonymous

payment method is needed. Indeed, this method should not reveal additional information about the group members to the service provider.

A. IBDT Signature Scheme

We outline a general identity-based dynamic threshold signature scheme, namely the second protocol proposed in [7]. Our protocol will be a slight modification of this general case; we will point out differences when needed. A general *IBDT* signature scheme consists of the following five probabilistic polynomial-time algorithms.

IBDT 1. ***Setup*** *is a randomized trusted setup algorithm that takes as input a security parameter λ, a universe of identities $\mathcal{ID}$ and an integer n which is the upper bound on the size of the threshold policies,* i.e. *the maximum number of users that can participate in a threshold signature. It outputs a set of public parameters* pms *and a master key pair* msk *and* mpk*. An execution of this algorithm is denoted as*

$$(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}) \leftarrow \mathsf{Setup}\,(\lambda, \mathcal{ID}, n)\,.$$

IBDT 2. ***Keygen*** *is a key extraction algorithm that takes as input the public parameters* pms*, the master key pair* msk *and* mpk*, and an identity $\mathbf{id} \in \mathcal{ID}$. The output is a private key $SK_{\mathbf{id}}$. An execution of this algorithm is denoted as*

$$SK_{id} \leftarrow \mathsf{Keygen}\,(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}, \mathbf{id})\,.$$

IBDT 3. ***Sign*** *is a randomized signing algorithm that takes as input the public parameters* pms*, the master public key* mpk*, a user's secret key $SK_{\mathbf{id}}$, a message $\mathsf{Msg} \in \{0,1\}^*$ and a threshold signing policy $\Gamma = (t, S)$ where $S \subset \mathcal{ID}$ and $1 \leq t \leq |S| \leq n$. Note that, in our case, t will be strictly equal to $|S|$. It outputs a partial signature $\sigma_{\mathbf{id}}$. We denote an execution of this algorithm as*

$$\sigma_{\mathbf{id}} \leftarrow \mathsf{Sign}\,(\mathsf{pms}, \mathsf{mpk}, SK_{\mathbf{id}}, \mathsf{Msg}, \Gamma)\,.$$

IBDT 4. ***Comb*** *is a deterministic signing algorithm which takes as input the public parameters* pms*, the master public key* mpk*, a message* Msg*, a threshold signing policy $\Gamma = (t, S)$ and the partial signatures of the set S_t, with $|S_t| \geq t$ (again, $|S_t| = t$ in our case), and outputs a global signature σ. We denote the action taken by the signing algorithm as*

$$\sigma \leftarrow \mathsf{Comb}\,(\mathsf{pms}, \mathsf{mpk}, SK_{\mathbf{id}}, \mathsf{Msg}, \Gamma, \{\sigma_{\mathbf{id}}\}_{\mathbf{id} \in S_t})\,.$$

IBDT 5. ***Verify*** *is a deterministic verification algorithm that takes as input the public parameters* pms*, a master public key* mpk*, a message* Msg*, a global signature σ and a threshold policy $\Gamma = (t, S)$. It outputs 1 if the signature is deemed valid and 0 otherwise. We denote an execution of this algorithm as*

$$b \leftarrow \mathsf{Verify}\,(\mathsf{pms}, \mathsf{mpk}, \mathsf{Msg}, \sigma, \Gamma)\,.$$

For correctness, for any $\lambda \in \mathbb{N}$, any integer $n \in poly(\lambda)$, any universe $\mathcal{ID}$, any set of public parameters and master key pair $(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk})$, and any threshold policy $\Gamma = (t, S)$ where $1 \leq t \leq |S|$, it is required that for

$$\sigma = \mathsf{Comb}\,(\mathsf{pms}, \mathsf{mpk}, SK_{\mathbf{id}}, \mathsf{Msg}, \Gamma, \{\sigma_{\mathbf{id}}\}_{\mathbf{id} \in S_t})\,,$$

$$\mathsf{Verify}\,(\mathsf{pms}, \mathsf{mpk}, \mathsf{Msg}, \sigma) = 1$$

whenever the values pms, mpk, msk have been obtained by properly executing the Setup algorithm, $|S_t| \geq t$, and for each $\mathbf{id} \in S_t$, $\sigma_{\mathbf{id}} \leftarrow \mathsf{Sign}(\mathsf{pms}, \mathsf{mpk}, SK_{\mathbf{id}}, \mathsf{Msg}, \Gamma)$ and $SK_{\mathbf{id}} \leftarrow \mathsf{Keygen}(\mathsf{pms}, \mathsf{mpk}, \mathsf{msk}, \mathbf{id})$.

B. Key Management

The anonymity provided by our accreditation method is a result of our key generation protocol and management solution. As we stated above, identity-based public key cryptosystems allow using arbitrary strings as public keys. In our protocol, every user U_i is given an ordered list of public keys that depend on some unique identifier of the user, such as his national identity card number, his phone number, the IMEI number of his phone or a combination of any of them. We will call this identifier $n_{U_i} = d_k^i d_{k-1}^i \dots d_1^i$, where d_j^i is the j-th last digit of n_{U_i} and typically ranges from 0 to 9.

To generate the list of public keys from an identifier n_{U_i}, we choose a value $\ell < k$ and take the ℓ last digits of n_{U_i}. This results in a vector of public keys

$$\mathbf{PK}_{U_i} = \left\{\mathsf{pk}_1^{d_1^i}, \dots, \mathsf{pk}_\ell^{d_\ell^i}\right\},$$

with every $\mathsf{pk}_j^{d_j^i}$ being an encoding of the digit and its position in n_{U_i}, for example:

$$\mathsf{pk}_j^{d_j^i} = j \,||\, d_j^i,$$

where $||$ is the concatenation operation. To illustrate this process, imagine $n_{U_i} = 12345678$ and $\ell = 4$. The resulting public key list would be

$$\mathbf{PK}_{U_i} = \{18, 27, 36, 45\}\,.$$

To prove the number of members in a group, the members will choose a common integer $j \in \{1, \dots, \ell\}$ so that the j-th public key in their list, *i.e.* $\mathsf{pk}_j^{d_j^i}$, is different for all of them. Then they will perform the required operations with these public keys and their corresponding private keys. Assuming that the values of the digits range from 0 to 9, this would provide anonymity to each of the users, since on average 10% of people will share the same public key $\mathsf{pk}_j^{d_j^i}$ for some value of j.

Note that this approach limits the size of the groups that can be certified with our method to a maximum of 10. Moreover, intuition tells us that the closer the size of the group to this maximum size, the more difficult it becomes to find a value of j for which each user has a different public key. The probability that our protocol fails depends on the number of keys each user is given, ℓ, and the size of the group n; more specifically for $n \leq 10$:

$$F(\ell, n) = \left(1 - \frac{10(10-1)\dots(10-n+1)}{10^n}\right)^\ell,$$

that is very close to 1 for values of n close to 10.

The limit on the maximum value of n can be increased by assigning $d \geq 2$ digits of n_{U_i} to each of the ℓ public keys,

instead of just one digit. By doing this, the maximum value for the size of the groups becomes 10^d, and the probability of failure, for values of $n \leq 10^d$, is

$$F(\ell, n, d) = \left(1 - \frac{10^d(10^d - 1)\dots(10^d - n + 1)}{10^{dn}}\right)^{\ell}.$$

However, the price to be paid for choosing a larger d is a loss of anonymity, since, if more digits are associated to each public key, less users share the same public key. For example, for $d = 2$ a user would share each of his keys with only 1% of the total number of users. The service provider will choose ℓ and d depending on maximum number of keys that a user can store, the maximum allowed group size and the anonymity level to be guaranteed.

C. Anonymous Payment Mechanisms

Group discounts are one of the applications of our method: after proving the group size, the group members must pay a fee that depends on that size. If proving the size has been done anonymously, it would be pointless to subsequently use a non-anonymous payment protocol (such as credit card, PayPal, etc.).

Hence, we need to use an anonymous payment mechanism along with our group size accreditation protocol. Electronic cash protocols such as [4] are good candidates for this role. Nowadays, Bitcoin [10] is a well-established electronic currency and, although it is not anonymous by design [11], it can be a good solution if accompanied by careful key management policies. Also, extensions of the original protocol as Zerocoin [9] provide anonymity by design.

In this work we propose a much simpler approach based on prepaid scratch cards that users can buy at stores. Each such card contain a code $\mathsf{Pay.Code}$ which the service provider will associate with a temporal account holding a fixed credit specified by the card denomination.

D. Communication Technologies

Our accreditation method requires communication among the members of a group and between the members and some type of verifying device. If we want to prove not only that a group has a certain number of members, but also that these are together, the interactions with the verifying device must rely on short-range communication technologies, like NFC, Bluetooth or WiFi.

During the accreditation protocol, the users' smartphones will be detected in some way by the verifying device and a communication channel will be established. The requirements and constraints of this process depend on the type of service and verifying devices, but nonetheless it is desirable that communication establishment be fast and not too cumbersome to the user.

We propose to use Bluetooth, and in particular *Bluetooth Smart* (BLE) [2] to communicate with the verifying device. BLE solves some of the main limitations of traditional Bluetooth, *i.e.* reduces detection and bonding times, requires much less work by the user than NFC and has a shorter range than both Bluetooth and WiFi, which is desirable in a method like ours. Finally, BLE is implemented by most major smartphone manufacturers, at least in recent models, unlike NFC.

Regarding communication between the smartphones, any of the three mentioned technologies, or a combination of them (*e.g.* Bluetooth pairing through NFC messages) seems appropriate. The choice is up to the service provider.

III. Group Size Accreditation Method

A service that implements our accreditation method includes the following elements:

- A service provider (SP) that publishes a smartphone application App_U and distributes the necessary public parameters and keys of an *IBDT* signature scheme Π to users, after some registration process.
- A smartphone application App_U for each user U which:
  - allows computing signatures with Π on behalf of U;
  - allows computing ciphertexts with a public-key encryption scheme Π' selected by SP, under SP's public key pk^{SP};
  - can be run on master or slave mode, which affects how App_U participates in the accreditation protocol.
  - includes some certificate which allows checking the validity of pk^{SP};
  - implements some communication protocol, relying in short-range communication technologies, such as NFC or Bluetooth, to interact with the applications of the rest of the members of the group and with the verifying devices.
- Prepaid payment scratch cards available at stores. Each card includes a code $\mathsf{Pay.Code}$ that the SP associates to an account with a fixed credit specified by the card denomination.
- Verifying devices installed at suitable places in the provider's infrastructures which:
  - allow verifying signatures with Π;
  - hold the SP certificates as well as the keys needed to decrypt ciphertexts produced with Π' under pk^{SP}.
  - have short-range communication capabilities and implement some protocol to communicate with the users' devices.
- Some method to penalize or prevent the misuse of the system.

The complete accreditation protocol runs as follows:

Protocol 1. ***System setup protocol.***

1) *SP chooses the user identifier to be used as n_U and appropriate values for ℓ and d.*
2) *SP generates the parameters of the* IBDT *signature scheme Π as per Algorithm* $\mathsf{IBDT.Setup}$*;*
3) *SP generates the parameters of the public-key encryption scheme Π'.*

Protocol 2. ***Registration protocol.***

1) *A user U with identifier n_U authenticates himself to the service provider, face-to-face or by some other means. The user receives a PIN code* pin_U.
2) *The service provider associates to U a vector of public keys of* Π, $\mathbf{PK_{id}}$ *as described in Section II-B.*
3) *The service provider computes the secret keys associated to* $\mathbf{PK_{id}}$ *as per Algorithm* IBDT.Keygen*:*

$$\mathbf{SK_{id}} = \left(sk_1^{d_1^{\mathbf{id}}}, \ldots, sk_\ell^{d_\ell^{\mathbf{id}}}\right).$$

4) *The user downloads the smartphone application* App_U *and, using the PIN code* pin_U*, completes the registration protocol and receives the system parameters and keys, as well as the public key* pk^{SP}.

Protocol 3. ***Credit purchase.***

1) *A user buys a prepaid card for the system,* e.g. *a scratch card, from a store.*
2) *The card includes some code* Pay.Code *which has to be introduced in the smartphone application.*

Protocol 4. ***Group setup protocol.***

1) *Some user U^*, among the group of users $U_1, \ldots, U_t$ who want to use the service, takes the leading role. This user will be responsible for most of the communication with the verifying device. U^* sets his smartphone application to run in master mode and the others set it to work in slave mode.*
2) *The users agree on a value $j \in \{1, 2, \ldots, \ell\}$ such that the value of the j-th public key in* $\mathbf{PK_{id}}$ *is different for every user.*

Protocol 5. ***Group size accreditation protocol.***

1) *A verifying device detects the users' devices and sends them a unique timestamped ticket* T *that may include a description of the service conditions and options.*
2) *Each user U_i runs Algorithm* IBDT.Sign *to compute a partial signature with* Π *under his secret key* $\mathsf{sk}_j^{d_j^i}$ *on message*

$$\mathsf{Msg} = \left\langle \mathsf{T} \,||\, \mathsf{pk}_j^{d_j^1} \,||\, \ldots \,||\, \mathsf{pk}_j^{d_j^t} \right\rangle,$$

for the threshold predicate $\Gamma = (t, \{\mathsf{pk}_j^{d_j^1}, \ldots, \mathsf{pk}_j^{d_j^t}\})$. It sends the resulting partial signature $\sigma_{\mathbf{i}}$ to U^*.
3) *U^* receives $(\sigma_1, \ldots, \sigma_t)$ and runs Algorithm* IBDT.Comb *to combine these signatures and output a final signature σ on behalf of $U_1, \ldots, U_t$. U^* sends to the verifying device*

$$\mathsf{Msg}' = \langle \mathsf{Msg}, \sigma \rangle.$$

4) *The verifier device checks the validity of the signature by running*

$$\mathsf{IBDT.Verify}(\mathsf{Msg}, \sigma, \mathsf{pk}_j^{d_j^1} || \ldots || \mathsf{pk}_j^{d_j^t}, t).$$

Note that this signature will only be valid if all users $U_1, \ldots, U_t$ have collaborated in computing it, and thus it proves that the group of users is composed of at least t people. If the signature is not valid, the group will be penalized in an application-dependent way, e.g. *with access denial, group discount denial, etc. Otherwise, the service provider grants access to the group of users and tells the group the amount* amount_t *they have to pay depending on the group size.*

Protocol 6. ***Payment.***

1) *Each group member U in the (sub)set P of group members who want to collaborate in paying the bill sends to the verifying device via Bluetooth or WiFi his payment code encrypted under SP's public key:*

$$C_U = \mathsf{Enc}_{\mathsf{pk}^{SP}}(\mathsf{T} || \mathsf{Pay.Code}_U),$$

where $\mathsf{Pay.Code}_U$ *is the code which user U obtained from a prepaid scratch card and where* Enc *is the public-key encryption algorithm of scheme Π'.*
2) *The verifying device decrypts the ciphertexts $\{C_U : U \in P\}$ to obtain the payment codes of the users in P.*
3) *The verifier device substracts the quantity* amount_t *divided by the cardinal of P to the accounts associated with the received payment codes.*

IV. Performance Analysis

Our group size accreditation method is to be run by service providers, specialized verifying devices and the users' smartphones. Therefore, it is important that the computations of the underlying cryptographic protocol be as fast as possible, especially the algorithms that are executed by the smartphones, which have limited computational capabilities and rely on batteries.

In this section, we analyze the performance of the underlying *IBDT* signature scheme. This scheme is a pairing-based cryptographic protocol and as such, the required operations are performed in elliptic curve groups. We analyze its performance by counting the number of point multiplications, point exponentiations and pairings, which are the most costly operations.

Table I shows the number of these operations for each of the algorithms in the *IBDT* signature scheme. The number of operations is counted as a function of the maximum number of possible participants in a signature, n, and the size of the signing group t. As we stated previously, $t \leq n$.

TABLE I
Operations required per algorithm

	Multiplications	Exponentiations	Pairings
Setup	0	$n+4$	1
Keygen	$2n$	$4n$	0
Sign	$2n+6$	$2n+5$	0
Comb	$2n-t+1$	$2n-t$	0
Verify	$n+2$	$n+1$	4

Note that the IBDT.Sign and IBDT.Comb algorithms, that are intended to be executed in the users' smartphones during the **group size accreditation protocol** (5), present what seems

to be a quite high number of operations. This might be a problem if the devices in which these algorithms are to be executed do not have enough computational power. Moreover, these two algorithms should precisely be most efficient, since they are run most often, and possibly with time constraints. Therefore, it would be interesting if we could precompute some of their operations.

The IBDT.Sign algorithm is a probabilistic protocol, that is, it has some random values in it that have to be refreshed each time it is executed. This limits the amount of operations in the algorithm that can be precomputed. On the other hand, most of the operations depend on static values, *e.g.* keys and threshold policies Γ. Threshold policies contain the number of signers that will participate in a signature and their public keys. We assume that groups of users will be quite stable, *i.e.* users will generally use services together with the same group members, or at least with a limited set of different groups. We can exploit this assumption by precomputing operations that only depend on static values and threshold policies.

The IBDT.Comb algorithm obviously depends on the output of IBDT.Sign, but it is a deterministic algorithm and some of its operations depend on static values and also on the threshold policies. Therefore, by the same assumption as before, we can precompute some of the operations.

These precomputations will divide the IBDT.Sign and IBDT.Comb algorithms in two phases each, one for precomputing values, which will be executed during the **group setup protocol** (Protocol 4), and the other one performed during the **group size accreditation protocol** (Protocol 5). The resulting number of operations in each of these phases is shown in Table II. It can be seen that the non-precomputable IBDT.Sign that needs to be run during the group size accreditation protocol involves a very small constant number of operations. The non-precomputable IBDT.Comb is not so light, but it is nonetheless lighter than IBDT.Comb without precomputation (Table I), because it no longer depends on the number n of users.

TABLE II
NUMBER OF PRECOMPUTABLE AND NON-PRECOMPUTABLE OPERATIONS FOR THE SIGN AND COMB ALGORITHMS.

	Multiplications	**Exponentiations**	**Pairings**
Prec. Sign	$2n+2$	$2n+1$	0
Non-prec. Sign	2	4	0
Prec. Comb	$2n-2t$	$2n-2t$	0
Non-prec. Comb	$3t+1$	$3t$	0

V. CONCLUSIONS AND FUTURE WORK

We have presented a privacy-preserving mechanism for group discounts. The method is built upon an *IBDT* signature scheme, a concrete key generation and management solution, short-range communication technologies and anonymous payment mechanisms. The complexity analysis shows that the method is usable in practice.

Future work will consist of developing a generic app for privacy-preserving group discounts that can be easily customized for specific applications.

ACKNOWLEDGMENTS AND DISCLAIMER

This work was partly funded by Google through a Faculty Research Award to the first author, who is also partially supported by the Government of Catalonia through an ICREA Acadèmia Prize. The following partial supports are also gratefully acknowledged: the Spanish Government under projects TIN2011-27076-C03-01 “CO-PRIVACY” and CONSOLIDER INGENIO 2010 CSD2007-00004 “ARES”, and the European Commission under FP7 projects “DwB” and “Inter-Trust”. The authors are with the UNESCO Chair in Data Privacy, but they are solely responsible for the views expressed in this paper, which neither necessarily reflect the position of UNESCO nor commit that organization.

REFERENCES

[1] G. R. Blakley, “Safeguarding cryptographic keys,” *Proceedings of the National Computer Conference*, pp.313–317, New York: AFIPS Press, 1979.
[2] Bluetooth SIG, “Specification of the Bluetooth System,” 2013. Available in https://www.bluetooth.org/en-us/specification/adopted-specifications.
[3] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” *Advances in Cryptology–CRYPTO 2001*, LNCS 2139, pp. 213–229, Springer, 2001.
[4] D. Chaum, A. Fiat and M. Naor, “Untraceable electronic cash,” *Advances in Cryptology–CRYPTO 88*, LNCS 403, pp. 319–327, Springer, 1990.
[5] A. De Caro and V. Iovino, “jPBC: Java pairing based cryptography,” *Computers and Communication (ISCC), 2011 IEEE Symposium on*, pp. 850–855, IEEE, 2011. Available in http://gas.dia.unisa.it/projects/jpbc/.
[6] J. Domingo-Ferrer, C. Ràfols and J. Aragonès-Vilella, *Method and system for customized contactless toll collection in carpool lanes* (in Spanish “Método y sistema de cobro sin contacto, por el uso de una vía, para vehículos de alta ocupación”), Spanish patent ref. no: P201200215. Date filed: February 28, 2012. Patent owner: Universitat Rovira i Virgili.
[7] J. Herranz, F. Laguillaumie, B. Libert and C. Ràfols, “Short attribute-based signatures for threshold predicates,” *Topics in Cryptology–CT-RSA 2012*, pp. 51–67, Springer, 2012.
[8] B. Lynn, *On the Implementation of Pairing-based Cryptosystems*, Doctoral dissertation, Stanford University, 2007.
[9] I. Miers, C. Garman, M. Green and A. D. Rubin, “Zerocoin: Anonymous distributed e-cash from bitcoin,” *Security and Privacy (SP), 2013 IEEE Symposium on*, pp. 397–411, IEEE, 2013.
[10] S. Nakamoto, “Bitcoin: A peer-to-peer electronic cash system,” *Consulted*, vol. 1, 2008. Available in http://www.bitcoin.org/bitcoin.pdf.
[11] F. Reid and M. Harrigan, “An analysis of anonymity in the Bitcoin system,” in *Security and Privacy in Social Networks*, eds. Y. Altshuler *et al.*, pp. 197–223, Springer, 2013.
[12] A. Shamir, “How to Share a Secret,” *Communications of the ACM*, 22:612–613, 1979.
[13] A. Shamir, “Identity based cryptosystems and signature schemes,” *Advances in Cryptology–CRYPTO 1984*, LNCS 196, pp. 47–53, Springer, 1985.

Autenticación No Interactiva para Internet de las Cosas

Francisco Martín-Fernández, Pino Caballero-Gil, Cándido Caballero-Gil
Departamento de Ingeniería Informática
Universidad de La Laguna
Emails: francisco.martin.07@ull.edu.es, pcaballe@ull.es, ccabgil@ull.es

***Resumen*—En este trabajo se propone un esquema de intercambio de información confidencial en entornos no seguros sobre redes móviles ad-hoc, basado en el concepto de demostración de conocimiento nulo no interactiva. De esta manera, se consigue que en una única comunicación se puedan inferir datos relevantes para la verificación de la legitimidad de los nodos de la red. Además, se propone el uso de este esquema aplicado a la autenticación y el control de accesos, a través del establecimiento de claves mediante la idea del protocolo criptográfico de Diffie-Hellman.**

***Palabras clave*—Autenticación (*Authentication*), Privacidad (*Privacy*), Demostración de Conocimiento Nulo (*Zero Knowledge Proof*), Internet de las Cosas (*Internet of Things*)**

I. Introducción

Cada vez es más frecuente ver cómo la tecnología se funde con la realidad en el uso cotidiano. A esta tendencia se le conoce como la Internet de las Cosas o IoT (Internet of Things) y surge ante la necesidad de tener monitorizado e interconectado cualquier dispositivo electrónico que sea útil para el ser humano. En esta nueva dimensión aparecen nuevos retos relacionados con la seguridad inalámbrica, ya que esta es la vía convencional de comunicación entre estos objetos hiperconectados. Se necesitan algoritmos criptográficos ligeros acordes a la capacidad reducida de cómputo de estos dispositivos. Gracias a la aparición de tecnología cada vez más potente y reducida en tamaño y peso, los esquemas criptográficos en comunicaciones inalámbricas han ido cambiando a pasos agigantados para adaptarse a esas nuevas condiciones. En particular, la evolución de las redes hacia la IoT ha conllevado que este proceso se acelere.

Con la aparición de este nuevo paradigma de objetos interconectados, donde la dimensión física se mimetiza con la dimensión lógica, será necesario codificar más de 100.000 millones de objetos, lo que equivaldría a que cada ser humano esté rodeado por 3000 objetos de media. Un aspecto fundamental a tener en cuenta es la forma de comunicación entre estos objetos. Debido al carácter móvil y al reducido tamaño de muchos de estos artilugios que conforman la Internet de las Cosas, esta comunicación debe ser inalámbrica. Además, la forma de agrupación a la que tienden según su naturaleza desemboca en que ese tipo de comunicación inalámbrica se establezca mediantes las denominadas redes móviles ad-hoc, también conocidas como MANETs (Mobile Ad-hoc NETworks). Estas redes están compuestas por dispositivos móviles, conectados inalámbricamente y generalmente se caracterizan por poseer algunas propiedades de auto-configuración.

Cada dispositivo que forma parte de una MANET posee libertad para desplazarse, lo que implica que las condiciones de enlace entre los dispositivos cambian dinámicamente y que cada nodo actúa como router de las comunicaciones ajenas. Otro aspecto relevante de la tipología de esta red es que en general puede operar de forma autónoma o bien estar conectada a Internet. Esta última posibilidad es muy útil en aquellas situaciones en las que los propios dispositivos no tengan una conexión directa a Internet.

En las MANETs existen muchos tipos de amenazas que podrían llegar a condicionar su uso. Una de las mayores amenazas es contra la seguridad de las comunicaciones mediante ataques de suplantación de identidad o escucha de la información enviada entre los nodos de la red.

Este trabajo propone el diseño de un nuevo esquema criptográfico ligero, en concordancia con la capacidad de cómputo de los nodos de la red, que permita asegurar las comunicaciones inalámbricas en una MANET.

Este trabajo se estructura en varias secciones. En la sección II se introducen brevemente algunos antecedentes. La sección III trata de forma pormenorizada un nuevo esquema de autenticación no interactiva. En la sección IV se explican posibles casos de uso alternativo para el esquema propuesto. Por último, en la sección V se detallan algunas conclusiones.

II. Antecendentes

En la bibliografía existen muchas propuestas [6], tanto basadas en criptografía simétrica [15] como en criptografía asimétrica [16]. La seguridad de muchos de los primeros esquemas es bastante fuerte, pero su mayor inconveniente es la distribución de la clave común entre los participantes en la comunicación. En un entorno como el de las MANETs [4] aplicadas a la Internet de las Cosas [1], presuponer la existencia de un canal totalmente seguro para transmitir claves simétricas sería una utopía. Además, el número de claves que se necesitaría sería demasiado elevado en una gran MANET basada sólo en criptografía simétrica. Precisamente para intentar subsanar este problema nació la criptografía asimétrica, también conocida como criptografía de clave pública. Además, la criptografía de clave pública permite firmar digitalmente la comunicación si primero el emisor cifra con su clave privada y luego el receptor descifra con la clave pública del emisor,

ya que así se consigue a la vez la identificación del remitente y la autenticación del mensaje. Curiosamente la capacidad de firma digital que otorga la criptografía de clave pública es lo que permite resolver el principal reto que presenta, que es la necesidad de establecer confianza en las claves públicas usadas. Para impedir posibles ataques MitM (Man in the Middle), se debe asegurar la identificación del usuario a quien corresponde cada clave pública. Existen diversos modelos para lograr esta certificación de claves públicas. El más habitual se basa en una infraestructura de clave pública o PKI (Public Key Infrastructure), que se basa en autoridades certificadoras. Otros esquemas se basan en una web de confianza. Una alternativa a las PKI es el uso de la criptografía basada en identidad en la que se hacen innecesarios los certificados.

El mayor inconveniente de los esquemas de criptografía asimétrica es su eficiencia computacional ya que en general los cálculos necesarios requieren bastante tiempo. Son sistemas, por lo general, demasiado pesados como para que funcionen con fluidez en entornos ligeros, como el de las MANETs en la IoT. Para subsanar este inconveniente se propone aquí la combinación de criptografía simétrica y de criptografía asimétrica mediante el uso de claves de sesión [5]. En particular, el modelo propuesto se basa en la generación de una clave de sesión compartida entre nodos previamente autenticados, con objeto de utilizar dicha clave de sesión para establecer comunicaciones secretas usando un sistema de criptografía simétrica.

Previo al establecimiento de las claves de sesión compartida se realiza una fase de autenticación de usuarios mediante un protocolo basado en la idea de las demostraciones de conocimiento nulo. Las demostraciones de conocimiento nulo o ZKP (Zero-Knowledge Proof) [10] establecen un método para probar el conocimiento de un secreto sin revelar ninguna pista sobre él.

En el ámbito de las MANETs usadas en IoT, una demostración de conocimiento nulo típica basada en sucesivos retos y respuestas implicaría un intercambio de sucesivos mensajes, lo que conllevaría tener que presuponer una conexión estable y continua entre los nodos. En entornos tan volátiles como la Internet de las Cosas, donde existen dispositivos que se pueden mover a gran velocidad (como por ejemplo, los vehículos que conforman las denominadas redes ad-hoc vehiculares o VANETs (Vehicular Ad-hoc NETworks), un intercambio masivo de mensajes para ejecutar una demostración de conocimiento nulo puede ser inviable debido a los posibles fallos de conexión durante el protocolo. Para subsanar el problema de la multitud de mensajes bidireccionales que producen las ZKP tradicionales han surgido en la bibliografía las demostraciones de conocimiento nula no interactivas [14], que condensan todos los retos en un único paquete enviado en un único mensaje. De esta forma el tiempo que conllevaría el intercambio de mensajes para llevar a cabo el protocolo interactivo se minimiza, de forma que sólo es necesario el envío de un único mensaje, pudiéndose incluso enviar este mensaje como beacon en modo broadcast a la red en la que se utilice el esquema.

III. Autenticación No Interactiva

Uno de los factores más importantes de los esquemas de demostración de conocimiento nulo, tanto interactivos como no interactivos, es la elección del problema matemático de base. En este trabajo en particular se utiliza el del isomorfismo de grafos. Un isomorfismo entre dos grafos se define mediante una biyección entre los conjuntos de sus vértices preservando la relación de adyacencia, o dicho de otro modo, cualquier par de vértices de un grafo son adyacentes si y solo si lo son sus imágenes en el otro grafo. El problema del isomorfismo de grafos consiste en determinar si dos grafos son isomorfos o no. Este problema ha sido utilizado en entornos criptográficos [12] [13] debido a que no se conoce un algoritmo eficiente para resolverlo en general. En particular, la determinación de si dos grafos con el mismo número de vértices v y de aristas a son isomorfos implicaría un ataque por fuerza bruta que exigiría comprobar si las $v!$ biyecciones posibles preservan la adyacencia. Curiosamente el problema del isomorfismo de grafos pertenece, en complejidad computacional, a la categoría NP, sin que se conozca hasta ahora si es resoluble en tiempo polinómico o bien si es NP-completo [11]. Por tanto su resolución, dependiendo del tamaño de los grafos implicados, puede ser muy costosa. Este problema permite crear varios compromisos a partir de un grafo original mediante sus posibles grafos isomorfos.

El esquema propuesto se basa en una variante de las demostraciones de conocimiento nulas no interactivas en la que sólo es necesario un único mensaje para poder verificar el conocimiento. La idea es conformar un sistema cuya seguridad pueda adaptarse según el nivel de seguridad que se requiera. De esta manera, cuanto más retos diferentes se consideren en la ejecución, más garantía tendrá el verificador. Concretamente los parámetros de la propuesta son:

- G: Grafo conocido por los nodos legítimos, sobre el que conocen una solución a un problema difícil.
- Sol_G: Solución al problema en G.
- $Reto_i$: Reto $i-$ésimo propuesto por el verificador.
- G_i: Grafo isomorfo $i-$ésimo usado como compromiso.
- $Iso(G, G_i)$: Isomorfismo entre G y G_i.
- $Res(Reto_i, G_i)$: Respuesta $i-$ésima correspondiente al $Reto_i$ sobre el grafo G_i.
- $h(\cdot)$: Función hash.
- $LSB(\cdot)$: Least Significant Bit o bit menos significativo de un string de entrada.
- $E_{k_i}(\cdot)$ Cifrado simétrico con clave k_i.
- $Subclave$: Contribución de un nodo a la clave de sesión.

Según la propuesta, el mensaje que cada nodo que desee autenticarse envía como nodo legítimo de la red está compuesto por una serie compromisos definidos mediante grafos isomorfos de un grafo conocido por todos los usuarios legítimos de la red. Por ejemplo, el grafo podría corresponderse con un grafo en el que los nodos representen a todos los usuarios de la red. En particular, esos compromisos se encuentran, todos salvo el primero, en principio cifrados de forma que se van descifrando a medida que se van verificando las respuestas anteriores.

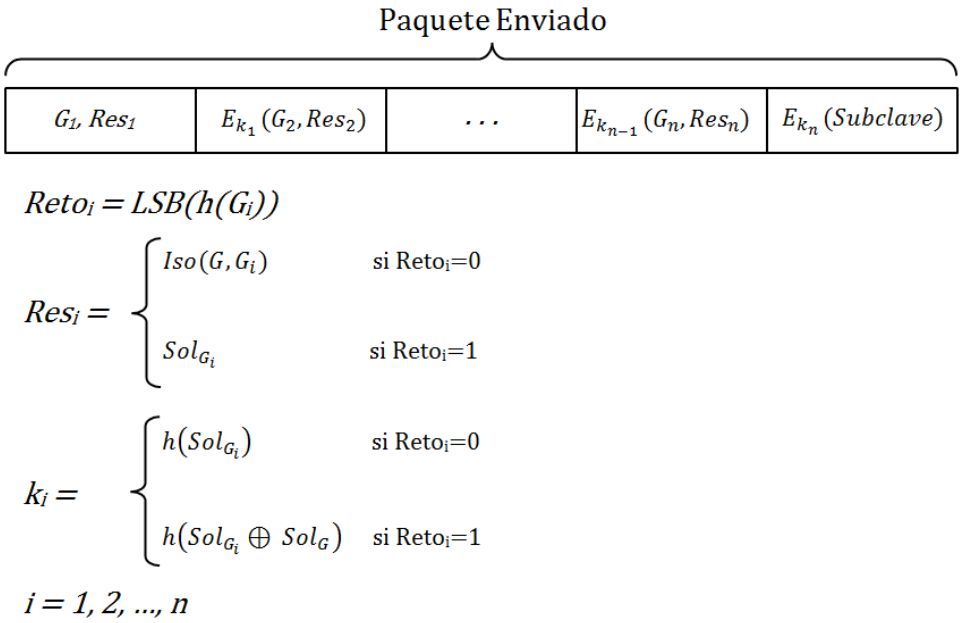


Figura 1. Componentes de los mensajes enviados según el esquema propuesto.

Concretamente el mensaje está dividido en segmentos cifrados con diferentes claves, exceptuando el primer segmento que está en claro (ver Figura 1). De esta manera, un usuario legítimo de la red puede autenticarse para unirse a una sesión si el verificador es capaz de descifrar todos los segmentos del mensaje y llegar a la parte de la contribución del otro nodo a la clave de sesión que se quiere compartir, que se esconde en el último segmento. Las claves de cifrado de cada segmento dependen del segmento anterior, de manera que aunque se pretenda descifrar únicamente el último segmento, es imposible ya que para ello se deben haber descifrado en cascada todos los segmentos anteriores. El nivel de seguridad del esquema depende del número de segmentos o retos que se incluyan en el mensaje ya que a mayor cantidad de segmentos, más complejo es llegar al último y obtener la información que se requiere para el establecimiento de la clave compartida. Tras la autenticación bidireccional siguiendo el mismo procedimiento, ambos nodos conocerán mediante un esquema del tipo Diffie-Hellman la clave de sesión compartida a partir de las dos subclaves intercambiadas.

Cada segmento contiene un grafo isomorfo del grafo original que conocen todos los usuarios legítimos del sistema. Además se establece una función hash unidireccional pública conocida por todos los nodos legítimos de la red. Por una parte esta función sirve para definir el reto que el usuario debe solucionar sobre cada grafo isomorfo de manera que sea conocido y estrictamente no maleable. Por otra parte, la función hash se utiliza en la definición de la clave de cifrado de cada segmento del mensaje.

El procedimiento de actuación del receptor del mensaje es:

1. Procesa el primer segmento del mensaje que está en claro.
2. Calcula, con la función hash, el reto que corresponde a la información que alberga el segmento.
3. Verifica si la respuesta corresponde al reto y grafo isomorfo.
4. A partir del reto calcula la clave que debe utilizar para descifrar el siguiente segmento.
5. Aplica los pasos del 2 al 4 hasta el último segmento, que una vez descifrado contiene la información necesaria para establecer la clave secreta compartida con el emisor.

Todos los usuarios legítimos de la red (ver Figura 2) poseen tanto el grafo original como una clave secreta correspondiente a dicho grafo, que es una solución a un problema difícil en ese grafo. Este podría ser por ejemplo un circuito hamiltoniano, ya que el problema del circuito hamiltoniano en un grafo arbitrario es NP-completo.

Como función hash, para discernir el reto correspondiente y calcular la clave de cifrado de cada segmento del mensaje se puede usar el nuevo estándar de función hash SHA-3 [2] [3].

En cuanto al cifrado simétrico para los distintos segmentos del mensaje, exceptuando el primero que se manda en claro, se puede optar por aplicar el cifrado en flujo usado en la cuarta generación de comunicaciones móviles (LTE o 4G) [8] [9], conocido como Snow3G [7], ya que entre sus características destaca una complejidad computacional lineal lo que garantiza eficiencia y rapidez en los procesos de cifrado y descifrado.

Como retos se han elegido los habituales aplicados a las demostraciones de conocimiento nulo basadas en grafos isomorfos. En el caso del esquema no interactivo planteado se definen los retos mediante el resultado de la función hash booleana aplicada sobre el grafo isomorfo compromiso. Para cada uno de los retos, la respuesta se define como sigue:

- $Reto = 0$: La respuesta es el isomorfismo.
- $Reto = 1$: La respuesta es la solución al problema en el

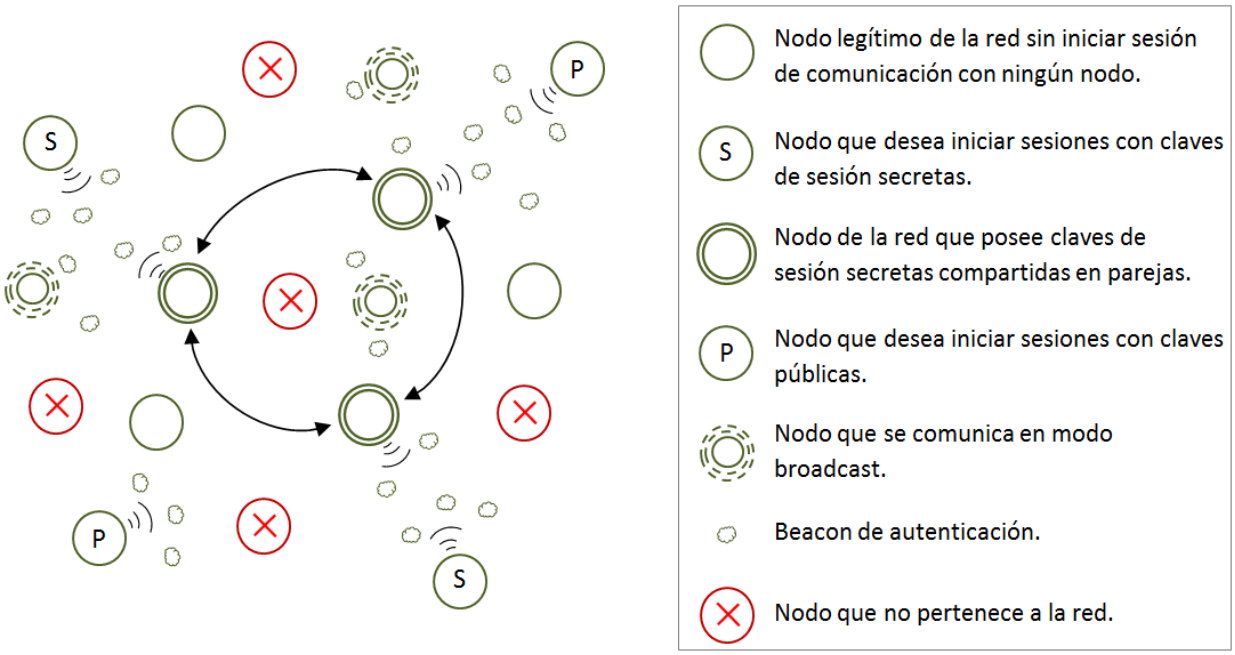


Figura 2. Tipos de nodos en el sistema propuesto.

grafo isomorfo.

A continuación se muestra el pseudocódigo de un posible algoritmo que debe ejecutar el receptor para implementar el esquema propuesto. Además en la figura 3 se muestra el diagrama de flujo de dicho algoritmo.

//Params: beacon, mensaje con segmentos cifrados
//Params: tseg, dimensión de los segmentos del beacon
//Params: solsec, solución al problema sobre el grafo G
//Return: Subclave, contribución a la clave obtenida del mensaje
function getDatos (char[] $beacon$, int tseg, char[] $solsec$)
01: var $segs$[]; // Almacena los segmentos del mensaje
02: // Se divide el mensaje en segmentos de tamaño $tseg$
03: $segs$ = $beacon$.splitByTam($tseg$);
04: // Se obtiene el grafo isomorfo y la respuesta en él
05: // Primer segmento que no está cifrado
06: var gi = getGi($segs$[0]);
07: var res = getRes($segs$[0]);
08: // Se calcula el reto a resolver
09: var $reto$ = LSB.hash(gi.getBytes());
10: // Comprueba que la respuesta es correcta para avanzar
11: **if** (res != respuesta(gi, $reto$))
12: return; // Si no es correcta se aborta
13: **endif**
14: // Obtiene la solución en gi
15: var sol = resolver(gi);
16: // ki es la clave de cifrado del segmento posterior
17: var $ki = \overline{reto} * hash(sol) \oplus reto * hash(sol \oplus solsec)$
18: var $descifrado$;
19: // Se repiten los siguientes pasos hasta el final
20: **for** (int $i = 1; i < segs.size() - 1; i++$){
21: // Se descifra el segmento con la clave ki
22: descifrado = Crypto.decrypt($segs[i]$, ki);
23: gi = getGi($descifrado$);
24: res = getRes($descifrado$);
25: $reto$ = LSB.hash(gi.getBytes());
26: **if** (res != respuesta(gi, $reto$))
27: return;
28: **endif**
29: sol = resolver(gi);
30: $ki = \overline{reto} * hash(sol) \oplus reto * hash(sol \oplus solsec)$
31: }
32: // El descifrado del último segmento sería
33: // la contribución a la clave compartida
34: return Crypto.decrypt($segs[segs.size()]$, ki);
endfunction

Una vez ejecutado el algoritmo descrito, sólo queda acceder al último segmento del mensaje y descifrarlo con la clave devuelta en la última iteración para poder obtener la contribución del nodo emisor a la clave de sesión compartida con cada uno de sus posibles interlocutores.

IV. Casos de Uso

Los principales casos de usos del esquema descrito son todos aquellos en los que se requiera el nivel de confidencialidad que otorgan las comunicaciones cifradas con claves de sesión secretas. De este modo, un caso de aplicación interesante podría ser el de las transacciones comerciales en MANETs. En este escenario, un nodo legítimo de la red quiere compartir un recurso propio con otros nodos legítimos para llevar a cabo una transacción comercial. Este recurso puede ser, por ejemplo, su acceso a Internet que desea compartir previo pago. Es frecuente que los nodos de una MANET, por su carácter móvil,

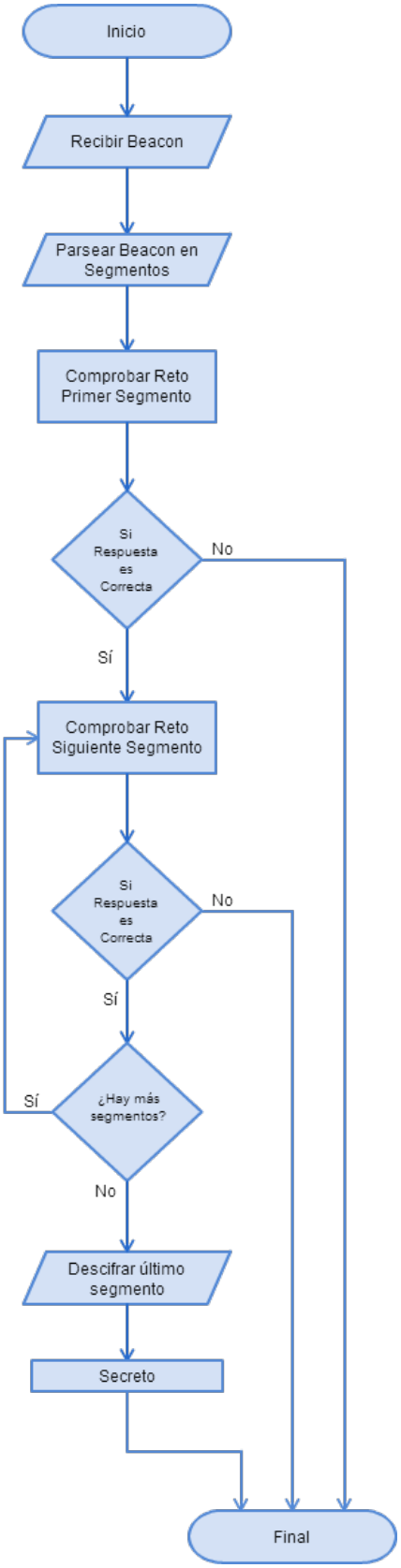


Figura 3. Diagrama de flujo del Algoritmo propuesto.

no posean acceso a Internet en muchos lugares. Un nodo legítimo de esa MANET que sí posea una conexión a Internet, puede tener como misión la comercialización de su conexión para que otros nodos legítimos hagan uso de ella. Esos nodos móviles que quieran hacer uso de esta conexión a la red de redes durante un tiempo limitado, sólo deben establecer una clave secreta de sesión compartida con el nodo emisor para comenzar las comunicaciones que le otorgan acceso a Internet. Para el establecimiento de las claves secretas de sesión se puede utilizar el esquema propuesto en este trabajo.

Otros dos escenarios diferentes para la utilización de dos variantes del esquema propuesto pueden ser para nodos legítimos de la red a los que sólo les interese notificar información de manera autenticada y unidireccional a otros nodos legítimos sin necesidad de usar claves secretas, y para nodos legítimos de la red que quieran compartir su clave pública de forma autenticada con otros nodos legítimos.

Por una parte, un nodo puede sólo desear hacer broadcast (ver Figura 2). De esta manera, otro nodo de la red que lo escuche, podrá fiarse de la información que quiere transmitir el nodo emisor ya que para confeccionar el beacon que se envía en modo broadcast es necesario conocer una clave secreta de red, que es utilizada para generar los grafos isomorfos y soluciones del protocolo descrito. Un ejemplo de caso de uso en este nuevo escenario es el de la notificación de eventos dentro de una VANET. Estos eventos que se envían en modo broadcast por parte de un usuario legítimo, pueden ser eventos de publicidad. Los distintos nodos de la red pueden recibir publicidad acerca de comercios que se encuentran cercanos a su ubicación y que también pertenezcan a la red. Para la retransmisión de esta publicidad se puede utilizar el esquema que se propone en este trabajo con el fin de que sólo los nodos legítimos de la red puedan enviar publicidad, evitando así el spam masivo de nodos que no pertenezcan a la red.

Por otra parte, un nodo puede querer anunciar su clave pública de forma autenticada sólo a aquellos nodos que también pertenezcan a la red. Para ello utilizará una variante del esquema propuesto (ver Figura 2), mediante la cual envía beacons periódicos que en su último segmento esconden la clave pública de ese nodo. Esto conlleva que sólo los usuarios legítimos de la red puedan acceder al último segmento del beacon, que contiene la clave pública del emisor, ya que los retos y respuestas están basadas en una clave secreta de red. La retransmisión de la clave pública en la que se basa este escenario puede servir para cualquier caso de uso de firma digital en MANETs ya que un usuario legítimo podrá enviar su clave pública a otros usuarios legítimos de la red de forma autenticada con objeto de posibilitar el uso de un esquema de firma digital en la MANET.

V. Conclusión

Con la proliferación de dispositivos electrónicos en múltiples ámbitos ha surgido un nuevo paradigma denominado Internet de las Cosas. Una de las mayores amenazas de un despliegue del tipo de redes que intervienen en IoT es la seguridad de las comunicaciones. Los objetos interconectados en IoT

suelen tener menos capacidad de cómputo que un ordenador convencional y sus comunicaciones suelen ser inalámbricas. Por este motivo se hacen necesarios nuevos algoritmos criptográficos ligeros que se adapten a estas características. En este trabajo, se presenta un nuevo esquema basado en la idea de las demostraciones de conocimiento nulo no interactivas en las que sólo es necesario el envío de un mensaje para compartir información confidencial. Como resultado del nuevo esquema propuesto se define su uso para el establecimiento autenticado de claves de sesión secretas entre pares de nodos legítimos de redes móviles ad-hoc. Además, el esquema que se ha diseñado puede ser utilizado por nodos que quieren enviar información autenticada en modo broadcast hacia otros nodos legítimos de la red. También se definen casos de uso para el intercambio autenticado de claves públicas en estas redes usando una variante del esquema que se propone en este trabajo. En definitiva, la nueva propuesta permite diseñar un nuevo protocolo basado en la idea de demostración de conocimiento nulo no interactivo en el que sólo es necesario el envío de un único mensaje en un sólo sentido.

Actualmente se está realizando la implementación del esquema propuesto en dispositivos móviles con objeto de analizar su comportamiento en entornos reales. Además su harán simulaciones en MANETs con diferentes configuraciones para estudiar la escalabilidad de la propuesta.

AGRADECIMIENTOS

Investigación financiada por el MINECO y la fundación FEDER mediante los proyectos TIN2011-25452 e IPT-2012-0585-370000, y la beca de investigación BES-2012-051817.

REFERENCIAS

[1] L. Atzori, A. Iera, G. Morabito, "The Internet of Things: A survey," *Computer Networks*, 2010.

[2] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, "Keccak sponge function family main document," *Updated submission to NIST*, Round 2, version 2.1, 2010.

[3] G. Bertoni, J. Daemen, M. Peeters, G. Van Assche, "The Keccak SHA-3 submission," *http://keccak.noekeon.org/Keccak-submission-3.pdf*

[4] P. Caballero-Gil, C. Caballero-Gil, J. Molina-Gil, C. Hernández-Goya, "Self-organized authentication architecture for Mobile Ad-hoc Networks," *WiOpt*, pp 217–224, 2008.

[5] C.L. Chen, C.T. Li, "Dynamic Session-Key Generation for Wireless Sensor Networks," *EURASIP Journal on Wireless Communications and Networking*, 2008.

[6] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, L. Uhsadel, "A Survey of Lightweight-Cryptography Implementations," *IEEE Design and Test of Computers*, vol. 4, no. 6, pp 522–533, 2007.

[7] P. Ekdahl, T. Johansson, "A New Version of the Stream Cipher SNOW," *Proceedings of SAC, LNCS 2595*, pp 37–46, 2003.

[8] P. Ekdahl, T. Johansson, "SNOW - a new stream cipher," *Proceedings of NESSIE Workshop*, 2000.

[9] ETSI/SAGE, "Specification of the 3GPP Confidentiality and Integrity Algorithms UEA2 and UIA2. Document 2," *SNOW 3G Specification*, version 1.1, Septiembre 2005.

[10] U. Feige, A. Fiat, A. Shamir, "Zero-knowledge proofs of identity," *Journal of Cryptology*, vol. 1, Issue 2 , pp 77–94, 1988.

[11] M.R. Garey, D.S. Johnson, "Computers and Intractability: A Guide the theory of NP-Completeness," *Freeman and Co.*, 1979.

[12] O. Goldreich, S. Micali, A. Wigderson, "Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems," *Journal of the ACM*, 38(3), pp 690-728, 1991.

[13] S. Goldwasser, S. Micali, C. Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on Computing*, 18(1), pp 186-208, 1989.

[14] J. Groth, "Short Non-interactive Zero-Knowledge Proofs," *Advances in Cryptology - ASIACRYPT 2010*, pp 341–358, 2010.

[15] A. Martin, "On Some Symmetric Lightweight Cryptographic Designs," *Doctoral Dissertion, PhD*, Supervisors: T. Johansson, M. Hell, 2012.

[16] M. Toorani, A. Beheshti, "LPKI - A Lightweight Public Key Infrastructure for the Mobile Environments," *IEEE Singapore International Conference on Communication Systems*, pp 162–165, 2008.

An Elliptic Curve Based Homomorphic Remote Voting System

M.A. Cerveró V. Mateu J.M. Miret F. Sebé J. Valera
Dept. Matemàtica, Universitat de Lleida. Jaume II, 69, 25001, Lleida, Spain.
{mcervero, vmateu, miret, fsebe, jvalera}@matematica.udl.cat

***Abstract*—A remote voting system allows participants to cast their ballots through the Internet. Remote voting systems based on the use of homomorphic public key cryptography have proven to be a good option for carrying out simple elections with a reduced amount of candidates. In this paper, we present a new system that makes use of the additive homomorphic capabilities of the Elliptic Curve ElGamal (EC-ElGamal) cryptosystem. All the stages of the system are described together with an experimental analysis section which provides an assessment on the type of election our system would be suitable for.**

***Index Terms*—Electronic Voting, Elliptic Curve Cryptography, Knapsack Problem**

I. Introduction

Electronic voting (*e-voting*) refers to the use of advanced technology to election processes. E-voting systems reduce the economic cost of an election and increase the speed and accuracy of vote tallying. An e-voting system allowing voters to cast their ballots through the Internet is called a *remote voting system*. The security provided by such a system should include, at least, the following features:

- *Authentication*: only people in the electoral roll can vote.
- *Unicity:* every participant can vote once at most.
- *Privacy:* votes can not be related to voter identities.
- *Fairness:* no partial results can be revealed before the end of the voting period.
- *Verifiability:* correctness of the process can be checked.
- *Uncoercibility:* nobody can prove that a voter voted in a particular way.

The previous security requirements are obtained by making use of advanced cryptographic techniques. Current remote voting systems can be classified into three main paradigms: *blind signature-based*, *mix-type* and *homomorphic tallying*.

In the *blind signature-based* paradigm [1]–[3], a voter authenticates against a trusted authority which is responsible for checking that the voter appears in the electoral roll and she has not voted before. In that case, voter's ballot (the encrypted vote) is blindly signed by that authority. The *polling station* only accepts ballots that have been properly signed by the authority. When the voting period is concluded, ballots are decrypted and tallied.

In the *mix-type* paradigm [4]–[9] a voter casts her ballot after having signed it. Once the voting period has ended, the polling station shuffles and re-encrypts (mixes) the collected ballots in order to break the relation between each ballot and the identity of the voter who cast it. After that, the mixed ballots are decrypted and tallied.

In *homomorphic tallying* schemes [10]–[15], participants cast their ballots encrypted under some public key cryptosystem having a homomorphic property. The received ballots are homomorphically aggregated by the polling station into a single or a set of ciphertexts whose decryption will show the result of the election. These systems require the votes to be coded in such a way that the final tally can be recovered from the cleartext of the aggregated ballots. Also, each voter has to prove in *zero-knowledge* that her ballot has been composed properly.

It is well known that homomorphic tallying systems do not scale well as the number of candidates increases. Despite their benefits regarding decryption (just one decryption is needed), homomorphic tallying systems need an additional decoding step in order to get the amount of votes for each candidate from an aggregated ballot cleartext. The method employed for coding votes should permit to get the election result at a reasonable processing time. It is also important to be able to manage a large enough amount of candidates and voters.

A. Contribution and Plan of this Paper

In homomorphic tallying remote voting systems, the ballots are encrypted using some public key cryptosystem. In elections with few candidates, the Elliptic Curve ElGamal (EC-ElGamal) cryptosystem turns out to be more efficient that ElGamal implemented over a multiplicative group. ElGamal requires 1024 bit long keys while EC-ElGamal achieves an equivalent security employing just 160 bits. Hence, EC-ElGamal provides better memory and computational costs. However, in elections with a large amount of candidates, EC-ElGamal becomes not as efficient as ElGamal.

In this paper, we present an e-voting system belonging to the homomorphic tallying paradigm based on the use of the Elliptic Curve ElGamal (EC-ElGamal) cryptosystem. Its vote coding system allows a large number of candidates while offering a good performance at the decoding step.

The paper is structured as follows: Section II presents some basic concepts of elliptic curve cryptography and the EC-ElGamal cryptosystem. Section III provides the description of of the proposed e-voting system, while Section IV emphasizes a special case: the Referendum. Then, Section V is dedicated to prove the security of the system and Sections VI and VII are devoted to experimental results, conclusions and future

work. Finally, Annex A presents the elliptic curves used in the experimental part of this work.

II. Preliminaries

An elliptic curve E defined over a finite field $\mathbb{F}_p$ is an equation of the form

$$E : y^2 = x^3 + ax + b, \tag{1}$$

with $4a^3 + 27b^2 \neq 0$. The set of points of the curve, denoted $E(\mathbb{F}_p)$, is composed of the points $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p$ satisfying Equation (1) together with the point at infinity, $\mathcal{O}$. The set $E(\mathbb{F}_p)$ can be endowed with an abelian group structure, having $\mathcal{O}$ as the identity element, by means of the chord-tangent method [16]. This method provides an operation for adding curve points. Given two curve points P and Q, the *elliptic curve discrete logarithm problem* (ECDLP) consists of finding an integer d satisfying $Q = d \cdot P$. The ECDLP is computationally hard when the cardinality of $E(\mathbb{F}_p)$ has a large prime factor. The assumed intractability of the ECDLP has led to the design of public key cryptosystems constructed over the group of points of an elliptic curve [17].

A. The Elliptic Curve ElGamal Cryptosystem

The *Elliptic Curve ElGamal* (EC-ElGamal) cryptosystem is composed of the following procedures.

1) *Set up:* A finite field $\mathbb{F}_p$ is first selected. After that, two integers a and b defining an elliptic curve E over $\mathbb{F}_p$ (see Eq. 1) are chosen so that the cardinality of $E(\mathbb{F}_p)$ has a large prime factor q. Finally, a point P of order q is taken as a generator of the order q cyclic subgroup of $E(\mathbb{F}_p)$. The values (p, E, P, q) are made public.
2) *Key generation:* Given the set up parameters, a private key is generated by randomly choosing an integer d in the range $[1, q-1]$. Next, its related public key Q is computed as $Q = d \cdot P$.
3) *Encryption:* A plaintext M consisting of a point of $E(\mathbb{F}_p)$ is encrypted under public key Q by computing

$$E_Q(M) = C = (A, B) = (r \cdot P,\ M + r \cdot Q), \tag{2}$$

where r is an integer selected randomly in the range $[1, q-1]$. Each encryption makes use of a different random r, whose value must be kept secret.
4) *Decryption:* If the private key d is known, a ciphertext C can be decrypted by computing

$$D_d(C) = B - d \cdot A.$$

The cleartext M is obtained as a result.

The EC-ElGamal cryptosystem has an homomorphic property. Let $C_1 = (A_1, B_1)$ and $C_2 = (A_2, B_2)$ be two ciphertexts encrypting M_1 and M_2, respectively. They are aggregated by computing,

$$C = C_1 + C_2 = (A_1 + A_2,\ B_1 + B_2).$$

The decryption of C will provide $M_1 + M_2$ as a result.

III. Our Proposal

The presented remote voting system is composed of the following parties:

- *Polling Station:* It coordinates the system. It is responsible for collecting the votes and publishing the final election result. When all the ballots have been collected and aggregated, it asks the *key storage trusted party* to decrypt the aggregated ballots. The received ballots are published on some publicly accessible bulletin board for verifiability purposes.
- *Participants:* They are voters able to emit a vote.
- *Key Storage Trusted Party (KSTP):* It is responsible for generating and storing the election private key and publishing the election public key material. When required, it will decrypt the ciphertexts containing the aggregation of cast ballots. The election public key should be certified to ensure its authenticity.

Next, the different stages of an election are explained in detail.

A. Set Up

Let us consider an election with m candidates. The collected ballots will be aggregated into packages containing n votes each.

Let t be an integer so that $t \leq m$ (for the sake of simplicity, we are assuming that $t \mid m$). The KSTP generates an elliptic curve E defined over a finite field $\mathbb{F}_p$ so that the cardinality of $E(\mathbb{F}_p)$ has t large prime factors. That is,

$$\#E(\mathbb{F}_p) = h \prod_{i=1}^{t} q_i,$$

with h being a small integer and each q_i being a large prime (at least 160 bits long). Next, an order $q = \prod_{i=1}^{t} q_i$ point P is chosen.

Since $\#E(\mathbb{F}_p)$ has t large prime factors, the group of points $E(\mathbb{F}_p)$ has t large cyclic subgroups. Next, a generator P_i for each subgroup is generated by computing,

$$P_i = \Big(\prod_{\substack{1 \leq j \leq t \\ j \neq i}} q_j \Big) \cdot P, \tag{3}$$

so that $\mathrm{ord}(P_i) = q_i$.

Then, the KSTP creates the election private key d by choosing its value randomly in $[1, q-1]$, computes the election public key $Q = d \cdot P$ and publishes all the previous parameters (the private key is stored privately in a safe place).

After that, the KSTP generates m points $\{M_1, \ldots, M_m\}$, each one representing a different candidate. Being $s = m/t$, the points are computed as shown in Table I. Note that points $M_1, \ldots, M_s$ are in the subgroup of $E(\mathbb{F}_p)$ generated by P_1, points are $M_{s+1}, \ldots, M_{2s}$ in the subgroup generated by P_2, and so on.

Table I
GENERATION OF A CURVE POINT FOR EACH CANDIDATE.

Base Point $\mathbf{P_1}$	**Base Point $\mathbf{P_2}$**	...	**Base Point $\mathbf{P_t}$**
$M_1{=}P_1$	$M_{s+1}{=}P_2$		$M_{(t-1)s+1}{=}P_t$
$M_2{=}(n{+}1){\cdot}P_1$	$M_{s+2}{=}(n{+}1){\cdot}P_2$		$M_{(t-1)s+2}{=}(n{+}1){\cdot}P_t$
...	...	...	...
$M_s{=}(n{+}1)^{s-1}{\cdot}P_1$	$M_{2s}{=}(n{+}1)^{s-1}{\cdot}P_2$		$M_{ts}{=}(n{+}1)^{s-1}{\cdot}P_t$
Candidates from 1 to s	Candidates from $s+1$ to $2s$		Candidates from $(t-1)s+1$ to $ts=m$

B. Voting

The voting process starts when a participant $\mathcal{P}$ wants to vote by electing a candidate. It is composed of four steps: *candidate choice*, *electoral roll checking*, *vote coding verification* and *vote packing*.

1) Candidate Choice: Let $M_{c_\mathcal{P}}$ be the curve point representing the choice of participant $\mathcal{P}$ who emits her vote by performing the following steps:

- Encrypt $M_{c_\mathcal{P}}$ under the public key Q. The encrypted vote $C_\mathcal{P} = E_Q(M_{c_\mathcal{P}})$ is generated as shown in Eq. (2).
- Compute the signature of the encrypted vote.
- Send $C_\mathcal{P}$ together with its signature to the polling station.

2) Electoral Roll Checking: When the polling station receives a ballot, it first checks the validity of its digital signature. Next, it checks that the voter who has cast it appears in the electoral roll and that she has not voted before. In that case, the voter is asked to prove that her ballot was correctly generated. This is done as described next.

3) Vote Coding Verification: The participant has to demonstrate that the cleartext of the ballot she is casting corresponds to a point representing one of the candidates. This demonstration is performed by means of a *zero knowledge proof*. It consists of a data exchange between the *Prover* (participant $\mathcal{P}$) and the *Verifier* (*polling station*). This proof does not leak any information about the actual choice of the voter. The following proof is an adaptation of the proof presented in [18] for its use on elliptic curve cryptography.

The *Prover* has to prove that her vote $C_\mathcal{P} = (A_\mathcal{P}, B_\mathcal{P}) = (r_\mathcal{P} \cdot P, M_{c_\mathcal{P}} + r_\mathcal{P} \cdot Q)$ is an encryption of one of the points in the set $\mathcal{M} = \{M_1, \ldots, M_m\}$ (the points of $E(\mathbb{F}_p)$ representing each of the candidates). In order to do that, the *Prover* generates the points A_k, B_k, for $1 \leqslant k \leqslant m$:

$$\begin{aligned} A_k &= w_k \cdot P + u_k \cdot A_\mathcal{P}, && \forall k \neq c_\mathcal{P};\\ A_{c_\mathcal{P}} &= s \cdot P,\\ B_k &= w_k \cdot Q + u_k \cdot (B_\mathcal{P} - M_k), && \forall k \neq c_\mathcal{P};\\ B_{c_\mathcal{P}} &= s \cdot Q, \end{aligned}$$

where $w_k, u_k, s \in [1, q-1]$, are random values. Next, the *Prover* computes

$$\begin{aligned} chall &= \mathcal{H}(A_1, A_2, \ldots, A_m, B_1, B_2, \ldots, B_m),\\ u_{c_\mathcal{P}} &= chall - \textstyle\sum_{k \neq c_\mathcal{P}} u_k,\\ w_{c_\mathcal{P}} &= s - u_{c_\mathcal{P}} r_\mathcal{P}, \end{aligned}$$

where $\mathcal{H}$ is some cryptographic hash function like SHA256 [19]. Finally, the *Prover* sends A_k, B_k, u_k, w_k for $1 \leq k \leq m$ to the *Verifier*.

The *Verifier* checks that

$$\begin{aligned} A_k &= w_k \cdot P + u_k \cdot A_\mathcal{P}, && \forall k \in [1, m],\\ B_k &= w_k \cdot Q + u_k \cdot (B_\mathcal{P} - M_k), && \forall k \in [1, m],\\ chall &= \textstyle\sum_{k=1}^{m} u_k, \end{aligned}$$

with $chall$ computed as $\mathcal{H}(A_1, A_2, \ldots, A_m, B_1, B_2, \ldots, B_m)$. This verification ensures that the voter has voted for a point in set $\mathcal{M}$. If all the checkings are satisfied, the signed ballot and the data required to verify it was properly generated are published on the bulletin board so that any external entity can check its correctness.

4) Vote Packing: The verified votes are homomorphically aggregated into packages. Each package is an aggregation of up to n ballots.

Let us consider a set of ballots $\mathcal{C} = \{C_{\mathcal{P}_1}, \ldots, C_{\mathcal{P}_n}\}$ that will be aggregated into package S_ℓ. Package S_ℓ is generated as:

$$S_\ell = \sum_{j=1}^{n} C_{\mathcal{P}_j}, \tag{4}$$

where $C_{\mathcal{P}_j} = \left(r_{\mathcal{P}_j} \cdot P,\ M_{c_{\mathcal{P}_j}} + r_{\mathcal{P}_j} \cdot Q\right)$. Hence, package S_ℓ is of the form,

$$S_\ell = \left(\sum_{j=1}^{n} r_{\mathcal{P}_j} \cdot P,\ \sum_{k=1}^{m} x_k \cdot M_k + \sum_{j=1}^{n} r_{\mathcal{P}_j} \cdot Q\right), \tag{5}$$

where x_k is the number of votes for the candidate k in this package and $\sum_{k=1}^{m} x_k = n$ is the capacity of the package.

C. Vote Opening

Once the election has finished, it is time to decrypt the ballots and tally the votes. This process can be divided into four steps: *decryption*, *unpacking*, *scrutiny* and *publication*.

1) Decryption: When the election is finished, the polling station has a set of aggregated packages that have to be decrypted. The KSTP is asked to decrypt them.

2) Unpacking: After decryption, the polling station has to obtain the amount of votes for each candidate from the cleartext of each package. A decrypted package is of the form $\widehat{S}_\ell = \sum_{k=1}^{m} x_k \cdot M_k$, or equivalently,

$$\widehat{S}_\ell = \sum_{i=1}^{t} \sum_{k=1}^{s} x_{k+(i-1)s} (n+1)^{k-1} \cdot P_i, \tag{6}$$

where $\sum_{k=1}^{m} x_k = n$ and $ts = m$.

$\widehat{S}_\ell$ is decoded as follows:

1) For each base point P_i, compute $z_i = \prod_{1 \leq j \leq t, j \neq i} q_j$:

$$\widehat{S}_{\ell,i} = z_i \cdot \widehat{S}_\ell = \sum_{k=1}^{s} x_{k+(i-1)s} (n+1)^{k-1} \cdot (z_i \cdot P_i)\,. \tag{7}$$

2) For each i, compute the values $x_{k+(i-1)s}$ for $1 \leq k \leq s$ which satisfy equation 7. This bounded discrete logarithm can be solved as a knapsack problem using the *Meet in the Middle* (MITM) algorithm as described next. In a preprocessing phase, compute $\sum_{k=s/2+1}^{s} x_k (n+1)^{k-1} \cdot (z_i \cdot P_i)$ for all the feasible combinations of x_k

values (those whose addition is not greater than n), and store each resulting point, together with the related x_k values in a hash table.
Also in a preprocessing phase, compute $\sum_{k=1}^{s/2} x_k(n+1)^{k-1} \cdot (z_i \cdot P_i)$ for each feasible combination of x_k values and store each result, together with the related x_k values, in an array.
At decoding, each point R in the array is taken and subtracted from $\widehat{S}_{\ell,i}$,

$$\widehat{S}'_{\ell,i} = \widehat{S}_{\ell,i} - R.$$

If $\widehat{S}'_{\ell,i}$ is in the hash table, we are done. In that case, the values $x_{k+(i-1)s}$ for $1 \leq k \leq s$ are obtained from the values stored together with R (in the array) and $\widehat{S}'_{\ell,i}$ (in the hash table). Notice that if $s = 1$, the algorithm can directly cast $\widehat{S}_{\ell,i}$ against the hash table.
The explained MITM algorithm achieves a good balance between computational cost and memory consumption. If no precomputed data were used, the required computing time would be too large. On the other hand, precomputing and storing all the feasible combinations would be unaffordable in terms of memory storage requirements.

3) Scrutiny: When each package $\widehat{S}_\ell$ has been decoded, the polling station adds all the votes to finally scrutiny the election result. The total amount of votes for candidate k is $\sum_{\ell=1}^{\mathcal{L}} x_k^\ell$, where $\mathcal{L}$ is the total number of packages and $\{x_1^\ell, \ldots, x_m^\ell\}$ are the values obtained from package $\widehat{S}_\ell$.

4) Publication: Finally, the election result is published on the bulletin board. At the end of the election, the bulletin board contains all the information needed to verify the correctness of the whole process. That is,

1) The result of the election (amount of votes for each candidate).
2) The electoral roll (name and public key of each participant).
3) The received ballots together with their digital signature and proof of correct composition.
4) Each aggregated package, S_ℓ, together with its cleartext $\widehat{S}_\ell$, and the amount of votes it contains for each candidate.

IV. Specific Case - Referendum

A *Referendum* is an election in which the voters can vote for *yes*, *no*, or *blank*. Next, we will show that such an election can be implemented very efficiently.

1) Set Up: We propose to choose an elliptic curve E over a finite field $\mathbb{F}_p$ whose group order has $t = 3$ large prime factors q_1, q_2, q_3. Hence, $\#E(\mathbb{F}_p) = h \cdot q_1 \cdot q_2 \cdot q_3$, with $q = q_1 \cdot q_2 \cdot q_3$ being a 480 bits long integer. Finally, we take an order q point P.

Since there are three possible options (candidates), we generate the following points: $P_1 = q_2 \cdot q_3 \cdot P$, $P_2 = q_1 \cdot q_3 \cdot P$, $P_3 = q_1 \cdot q_2 \cdot P$, satisfying that $ord(P_i) = q_i$. This way, we can code each option in a different base point, so that $s = 1$. Table II shows the three options represented by those base points.

Table II
The points representing the options in a *Referendum*.

Option ***Yes***	Option ***No***	Option ***Blank***
$M_1 = P_1$	$M_2 = P_2$	$M_3 = P_3$

2) Unpacking: Since $s = 1$, the unpacking process can be carried out in a very fast way. In the preprocessing phase of the MITM algorithm, we store all the possible values for each option $\{0 \cdot P_i, 1 \cdot P_i, \ldots, n \cdot P_i\}$ in the hash table so that the unpacking operation for each choice can be solved through a single hash lookup.

V. Security Analysis

In this section we show how the proposed system achieves the security requirements enumerated in Section I.

1) Authentication: Each ballot is digitally signed by the participant who casts it. Hence, the polling station can authenticate the voter and check that she appears in the electoral roll. Moreover, the electoral roll and the received ballots are made publicly available on the bulletin board so that any entity can check that all the ballots have been cast by an authenticated participant.

2) Unicity: Unicity is composed of two requirements:

- The system must ensure that every voter votes only once.
- The system must ensure that each ballot contains only one vote. That is, a ballot can only encrypt a single point of list $\mathcal{M}$.

The first item is addressed by keeping a register of the voters that have already voted. If any participant tried to cast two or more ballots, the system would only accept the first one. The second item is ensured by means of the zero knowledge proof of ballot correct composition (see Section III-B3).

3) Privacy: Privacy of the choice made by a participant holds on the following facts:

- All the votes are encrypted using the EC-ElGamal cryptosystem, so that no information can be obtained from an encrypted ballot.
- All the encrypted votes are homomorphically packed, and only the resulting packages are decrypted. As a result, the voter and her choice are decoupled.
- Only the aggregated packages are decrypted. This is achieved by considering the KSTP is a trusted party which acts honestly.
- The proof needed to ensure that a ballot was correctly composed is *zero knowledge*. Hence, no information leaks from it.

4) Fairness: Assuming a correct praxis of the KSTP, no vote is decrypted before the opening stage, which takes place after the ending of the voting period.

5) Verifiability: The verifiability of our system is based in four points:

- The electoral roll is public and all the received (signed) ballots are also made public. Hence, any entity can check all the ballots come from an authenticated participant.
- Each *zero knowledge proof* of correct ballot composition is published on the bulletin board so that it can be checked by any entity which will get convinced that each ballot is coding a single vote.
- The homomorphic packing operation can be performed by any entity and next check that the obtained packages correspond to those published on the bulletin board.
- The decryption carried out by the KSTP can be performed verifiably [20].

Our proposal offers end-to-end verifiability: the correctness of the whole process can be verified by everyone.

6) Uncoercibility: Uncoercibility can be provided by applying any coercion-resistance solution like [21].

VI. Experimental Results

The most time consuming part of the proposed system is given by the unpacking step. Hence, we have developed a test program to check the time and memory consumption of the MITM algorithm proposed for solving that step. The program has been implemented in *C++* using the library *Crypto++* and has been executed in a PC with an *Intel Core i5 650 3.2GHz* CPU with 6GB of RAM running *Debian 8.0 Jessie* as OS. Table III shows the data extracted during the tests using elliptic curves with 160, 320 and 480 bits long cardinalities (the used elliptic curves are shown in Annex A). The columns *Preprocess time* and *Memory* concern to the time and memory consumption of an unpacking operation.

Table III
Time and memory consumption using 160, 320 and 480 bits elliptic curves with packages of 200 votes.

EC (bits)	#Base Pnts.	#Cands. Base Pnt.	Preprocess time (s)	Unpacking time (s)	Memory (MB)
160	1	4	4.474	0.089	4.337
		5	225.768	0.081	148.895
320	2	4	81.920	0.196	8.674
		5	4 254.340	0.175	297.791
480	3	4	243.798	0.286	13.010
		5	12 571.200	0.251	446.686

As we can see, the most time consuming part corresponds to the generation of the preprocessed data, which can be performed some days before the election takes place. This preprocessed data permits to unpack packages at a very reduced time. This last operation has to be fast because it determines the delay between the end of the voting period and the publication of the results. Table III shows that the proposed system is able to unpack packages with 200 votes and 15 candidates very efficiently (see the last row, corresponding to 5 candidates per base point). Furthermore, the memory needed to store the data generated during the preprocessing has a reasonable size which can be perfectly stored by any commodity PC.

We have also analyzed the time and memory consumption in the Referendum case, which requires the use of an elliptic curve with a 480 bits long cardinality and 3 cyclic subgroups. Table IV highlights the efficiency of our system to resolve elections with 3 candidates, needing a little more than 3 milliseconds to unpack a 200 votes package. The memory requirements and preprocessing time are negligible.

Table IV
Time and memory consumption for the *Referendum* case with packages of 200 votes.

Preprocess time (s)	Unpacking time (s)	Memory (MB)
0.599	0.003	0.064

Furthermore, our system can deal with larger packets. Table V shows the time and memory consumption when working with packets aggregating 1000000 votes. Although the preprocessing time and the memory requirements increase, the time required for unpacking keeps constant for any package size.

Table V
Time and memory consumption for the *Referendum* with packages of 1000000 votes.

Preprocess time (s)	Unpacking time (s)	Memory (MB)
3 395.550	0.003	320.435

By comparing our referendum system with that presented by Peng et al. [15], we can see that our unpacking algorithm is much more efficient than that in [15] (also implemented in *C++* using *Crypto++* and executed in the same PC). The proposal in [15] is implemented using the multiplicative homomorphic property of ElGamal cryptosystem over a multiplicative group. When using a 1024 bits public key, it can only manage packages of up to 440 votes, while our proposal can manage much bigger packages, as it can be seen in Table V. Moreover, the system in [15] requires 0.022 seconds to decode a 440 votes package while our system can perform an equivalent operation employing only 0.003 seconds.

VII. Conclusion and Future Work

A new e-voting system that makes use of the EC-ElGamal cryptosystem has been proposed. The new proposal makes use of a MITM algorithm to unpack aggregated ballots at a high speed. Our system can be used in an election with a large amount of candidates. The experiments carried out have shown that our proposal is faster than the multiplicative homomorphic ElGamal cryptosystem proposed by Peng et al. [15], in the case of referendum type elections.

As future research, we will investigate techniques to further reduce the time devoted to ballot unpacking.

Acknowledgments

Research of the authors was supported in part by grants MTM2010-21580-C02-01 (Spanish Ministerio de Ciencia e

Innovación), 2014SGR-1666 (Generalitat de Catalunya) and IPT-2012-0603-430000 (Spanish Ministerio de Economía y Competitividad).

APPENDIX

In this appendix we show the elliptic curves used in our experiments. Finding elliptic curves with a given cardinality is a hot topic of research. Several algorithms have been proposed to this end. The most widely known was proposed by Atkin and Morain [22] but there exist other approaches like that proposed by Agashe et al. [23], or that by Bröker et al. [24], [25]. In particular, the curves used in this paper have been generated using the algorithm described in [25].

Elliptic Curve with a 160 bits Cardinality

Prime number p**:** 1461501637330902918203684832716283019655932542983
Coefficient a**:** 1268133167195989090596625406312984755854486256116
Coefficient b**:** 386736940269827655214118852806596527602892573734
$\#E(\mathbb{F}_p)$**:** 1461501637330902918203684149283858612734394057783

Elliptic Curve with a 320 bits Cardinality

Prime number p**:** 5339967589802275205987554265423880286506761306\\
0392702776566091642653545140104649919593717477026 17
Coefficient a**:** 2088105959680623325842477250435045284830027862785\\
8702114334928250967380570135558516700552686822 13
Coefficient b**:** 1163755670441028554302599764553789716846705580467\\
5298547896235140718783999833532886192612103381 91
$\#E(\mathbb{F}_p)$**:** $2 \cdot 1461501637330902918203684832716283019655932542983 \cdot$
$\cdot 1826877046663628647754606040895353774569915678761$

Elliptic Curve with a 480 bits Cardinality

Prime number p**:** 9950573504132225237528841169965717595736565 8\\
0696044268676583193632704892755795210303651532931 5\\
9676515285332084476809492997893652227166881856911 3
Coefficient a**:** 2707069958412506902602824447813694027990688183 95\\
6223888320767933486020623060222427867360955451 04\\
4640503732397723488780961683652364955407331149336
Coefficient b**:** 4536433587307211432323192274338329543592280212 78\\
3596492035047326660663646635350354568929622029 50\\
7484145388525397790533006611631792360854156270441
$\#E(\mathbb{F}_p)$**:** $2 \cdot 1461501637330902918203684832716283019655932542983 \cdot$
$\cdot 1552845489664084350591415134761050708384428327041 \cdot$
$\cdot 2192252455996354377305527249074424529483898814481$

REFERENCES

[1] D. Chaum, "Security without identification: transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.

[2] A. Fujioka, T. Okamoto, and K. Ohta, "A practical secret voting scheme for large scale elections," in *Advances in Cryptology – AUSCRYPT '92*, ser. LNCS, vol. 718, 1993, pp. 244–251.

[3] M. Ohkubo, F. Miura, M. Abe, A. Fujioka, and T. Okamoto, "An improvement on a practical secret voting scheme," in *ISW '99*, ser. LNCS, vol. 1729, 1999, pp. 225–234.

[4] K. Sako and J. Kilian, "Receipt-free mix-type voting scheme: A practical solution to the implementation of a voting booth," in *Advances in Cryptology – EUROCRYPT '95*, ser. LNCS, vol. 921, 1995, pp. 393–403.

[5] M. Jakobsson, "A practical mix," in *Advances in Cryptology – EUROCRYPT '98*, ser. LNCS, vol. 1403, 1998, pp. 448–461.

[6] F. Sebé, J. M. Miret, J. Pujolàs, and J. Puiggalí, "Simple and efficient hash-based verifiable mixing for remote electronic voting," *Computer Communications*, vol. 33, no. 6, pp. 667–675, 2010.

[7] J. Puiggalí and S. Guasch, "Eficiencia y privacidad en una mixnet universalmente verificable," in *XI Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*, 2010, pp. 159–164.

[8] K. Peng, "An efficient shuffling based eVoting scheme," *J. Syst. Softw.*, vol. 84, no. 6, pp. 906–922, 2011.

[9] V. Mateu, J. M. Miret, and F. Sebé, "Verifiable encrypted redundancy for mix-type remote electronic voting," in *EGOVIS 2011*, ser. LNCS, vol. 6866, 2011, pp. 370–385.

[10] J. D. Cohen and M. J. Fischer, "A robust and verifiable cryptographically secure election scheme," in *26th Annual Symposium on Foundations of Computer Science*, 1985, pp. 372–382.

[11] K. Sako and J. Kilian, "Secure voting using partially compatible homomorphisms," in *Advances in Cryptology – CRYPTO '94*, ser. LNCS, vol. 839, 1994, pp. 411–424.

[12] R. Cramer, R. Gennaro, and B. Schoenmakers, "A secure and optimally efficient multi-authority election scheme," in *Advances in Cryptology – EUROCRYPT '97*, ser. LNCS, vol. 1233, 1997, pp. 103–118.

[13] M. Hirt and K. Sako, "Efficient receipt-free voting based on homomorphic encryption," in *Advances in Cryptology – EUROCRYPT 2000*, ser. LNCS, vol. 1807, 2000, pp. 539–556.

[14] K. Peng, R. Aditya, C. Boyd, E. Dawson, and B. Lee, "Multiplicative homomorphic e-voting," in *INDOCRYPT 2004*, ser. LNCS, vol. 3348, 2004, pp. 61–72.

[15] K. Peng and F. Bao, "Efficient multiplicative homomorphic e-voting," in *ISC 2010*, ser. LNCS, vol. 6531, 2011, pp. 381–393.

[16] J. H. Silverman, *The Arithmetic of Elliptic Curves*. Springer-Verlag, 1986.

[17] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Springer-Verlag, 2004.

[18] R. Cramer, I. Damgård, and B. Schoenmakers, "Proofs of partial knowledge and simplified design of witness hiding protocols," in *Advances in Cryptology – CRYPTO '94*, ser. LNCS, vol. 839, 1994, pp. 174–187.

[19] "FIPS 180-2: Secure Hash Standard," NIST, 2002.

[20] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *Advances in Cryptology – CRYPTO '92*, ser. LNCS, vol. 740, 1993, pp. 89–105.

[21] A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," in *WPES '05*, 2005, pp. 61–70.

[22] A. O. L. Atkin and F. Morain, "Elliptic curves and primality proving," *Math. Comput.*, vol. 61, no. 203, pp. 29–68, 1993.

[23] A. Agashe, K. Lauter, and R. Venkatesan, "Constructing elliptic curves with a given number of points over a finite field," Cryptology ePrint Archive, Report 2001/096.

[24] R. Bröker and P. Stevenhagen, "Elliptic curves with a given number of points," in *ANTS-VI*, ser. LNCS, vol. 3076, 2004, pp. 117–131.

[25] R. Bröker and P. Stevenhagen, "Efficient CM-constructions of elliptic curves over finite fields," *Math. Comput.*, vol. 76, no. 260, pp. 2161–2179, 2007.

On the revocation of malicious users in anonymous and non-traceable VANETs

Cándido Caballero-Gil,
Jezabel Molina-Gil
Departamento de Estadística,
Investigación Operativa y Computación
Universidad de La Laguna
Email: {ccabgil|jmmolina}@ull.es

Juan Hernández-Serrano,
Olga León,
Miguel Soriano-Ibañez
Departamento de Ingeniería Telemática,
Universitat Politècnica de Catalunya (UPC)
Email: {jserrano|olga|soriano}@entel.upc.edu

***Abstract*—For the proper performance of Vehicular Ad-hoc NETworks (VANETs) it is essential to protect the service against malicious nodes aiming at disrupting the proper operation by injecting fake, invalid data into the network. It is common to define a traditional identity-based authentication for vehicles, which are loaded with individual credentials. However, the use of these credentials in VANETs may allow vehicle tracking and thus violate users' privacy, a risk that can be overcome by means of anonymity schemes. This comes at the expenses, however, of on the one hand preventing VANET authorities from identifying malicious users and revoking them from the network, or on the other hand to avoid anonymity of users in front of the CA thus to allow their revocation. In this work, we describe a novel revocation scheme that is able to track and revoke specific users only after a number of complaints have been received while otherwise guaranteeing vehicle's anonymity.**

***Index Terms*—VANET, revocation, k-anonymity**

I. Introduction

Nowadays, there is a bunch of GPS-based applications offering traffic services based on information provided by local road authorities, police departments and systems that track traffic flow. Some of these applications, such as Google Traffic [7], TomTom [16], Sygic [15] or Waze [18], can even provide near real-time data about traffic status and congestions. However, for these services to properly work, users should provide information with at least their location to the companies offering these services without any guarantee that these companies will use this data for other considerations [1]. Therefore, users' privacy may be at risk.

Conventional VANETs and current traffic applications do not protect users' privacy. They can breach the privacy of the user of the vehicle because they manage information that reveals the location of vehicles at every moment. Nodes and/or users' privacy may be violated by the Certification Authority (CA) as long as it provides their certificates, but also by companies managing traffic data or Traffic Authorities (TAs), which can locate and track vehicles based on their transmissions. Protecting users against tracking can be solved by providing user anonymity but, at the same time, this lack of tracking avoids the revocation of malicious/misbehaving nodes disrupting the service operation.

In this work we propose a novel scheme that protects users' privacy in front of other users, TAs and even the CA while also offering the possibility to track malicious user and thus to throw them out of the system when a predefined amount of complaints have been received. As later explained malicious user can only be tracked after a predefined amount of complaints are received by the CA. To do this we will use k-anonymity protection that allow that the information for nodes contained in the release cannot be distinguished from at least k-1 individuals whose information also appears in the release.

This paper is organized as follows. Section II covers related research about privacy in VANETs. In Section III we present the proposed method to guarantee anonymity of users while allowing tracking malicious users. In section IV we derive an analytical model to analyze the efficiency of the method. This model has been validated via simulation in section V, where we also provide the results obtained by means of the model for a real scenario. Finally, the conclusions of this work are presented in Section VI.

II. Related Research

Sweeney proposed k-anonymity at first in 2002 [14] and its original intention was to thwart the ability to link field-structured databases, but has been viewed more broadly, and have been applied to many other fields, such as VANET.

In [13], authors use k-anonymity in VANET applications where k-anonymity is provided by a centralized *anonymizer* based on the users' real location. The author proposes a homomorphism for the location of a group of users that are near to others users. However, tracking with less precision is already possible and users need to wait until at least $k-1$ other users are close to their location to achieve enough anonymity. This delay reduces the quality of the users localization in time and space which compromises real-time service availability and accuracy. Thus, this approach does not work in case of real-time services and also in low density areas of users.

[19] proposes a hybrid and social-aware location-privacy in Opportunistic mobile social networks (HSLPO), a collaborative and distributed obfuscation protocol that offers location-privacy k-anonymity.

In [2] authors propose a self-managed VANET without CA based on Certificates Graphs where every node has a pseudonym and many sub-pseudonyms that change frequently in a range of time, at this way, passive users cannot track other users. In this scenario there is neither a RSU nor any cloud connection. Therefore, tracking from the cloud is impossible. The unique way to track another user is physically because the user must be authenticated with another user to track it.

In [20] vehicles entering a group can anonymously broadcast vehicle-to-vehicle (V2V) messages, this is another way to preserve privacy but the TM has the ability to retrieve the real identity of dishonest vehicles that are sending fake messages to other vehicles to disrupt traffic, so the privacy is violated.

[4] present different privacy-preserving variants to ensure that vehicles volunteering to generate and/or endorse trustworthy announcements do not have to sacrifice their privacy.

[11] proposes a protocol to exclude malicious network nodes based on complaints received from other vehicles. [17] presents another protocol that uses decentralized revocation voting. [12], [8], [9] are other approaches for conditional anonymity in VANET.

To the best of our knowledge, none of the proposals in the literature provide both complete anonymity (even against the CA providing the credentials) while allowing to later identify an anonymous user in order to revoke him/her from the system. In our proposal we achieve this goal only and only if several complaints against a user is received.

III. A TRACEABLE K-ANONYMITY METHOD FOR VANETS

In this section, we propose a method that provides k-anonymity [14] in VANETs while still guaranteeing that malicious users will be traceable. The method operation is as follows. Every user is randomly associated to a group n with k members that share cryptographic material, i.e., a pair of private-public keys (PKu_{G_n}, PKs_{G_n}), and a group certificate $Cert(G_n, t)$, which will be used to sign messages and authenticate data. In order to reduce the computational cost, we assume the use of cryptography based on elliptic curves [3]. We also consider the existence of a CA, which is responsible for creating the groups, maintaining a database with the group membership, distributing the cryptographic material among users and revoking users that misbehave.

When a user detects an undesired behavior from another user, such as the injection of false data, it presents a complaint to the CA. Such complaint is signed with its group key and must report the group identifier of the malicious user. In its turn, the CA flags all the users belonging to that group and stores this information at its database. As users do not reveal their particular identities but only their group identifier, both the malicious user and the one sending the complaint cannot be distinguished from other users belonging to the same group. Thus, they cannot be tracked by the CA or by other users. This feature protects users' privacy, but it makes a hard task to revoke and isolate malicious users from the network.

In order to achieve the traceability of malicious nodes to revoke them, we propose the use of group certificates with short-term expiration dates that henceforth we will call $roundr$. When the group certificate of a user is about to expire it must send a query to the CA to update it, which will check if the user is revoked based on the number of flags it has received. The rationale behind such mechanism is that users will change of group over time and will be flagged whenever they belong to the group reported in a complaint. Due to the fact that you can not identify what node is having a bad behaviour, only one complaint is valid in each round. Assuming that malicious users repeatedly misbehave, they will be flagged at least the same amount of times or more than any other honest node in the network and thus they can be easily revoked by the CA. Note that, a revoked user will be expelled from the system once it has to update its certificate, as it will not be able to acquire a new valid certificate.

The certificate updating process is a key point for the VANET safety and operation. On the one hand, if the time for updating the certificate t is short, it greatly increases processing data on the server and number of server-users communications. On the other hand, if this measure is too long, malicious users would remain in the network for a longer time without being revoked. As a first, trivial approach, given that the average trip time by car is about 22 minutes each way [5], the lifetime of a certificate t should be no more than $\frac{22*2}{f}$, with f the number of complaints that the CA needs to revoke a user if we want to remove malicious users from the system in a single day. Furthermore, in order to reduce overhead, multiple certificates can be issued without interaction between CA and the user [10].

The level of anonymity provided by this method increases as the group size k does. However, as previously mentioned, k-anonymity complicate the process of revoking users. In section IV we derive an analytical expression for the number of false positives and false negatives, i.e., the number of honest users being revoked and the number of malicious users that remain in the network, provided by this method as a function of the anonymity value k, the number of complaints needed to revoke users and the number of group changes.

The CA should guarantee that the assignment of a user to a given group is kept secret and exclusively known to the user and itself, and that only the members of the group have access to the group cryptographic material. Because of this, every user should be provided with a public/private key pair and the corresponding certificate when entering the network. Such cryptographic material would be exclusively used for communication with the CA in order to authenticate the user against it and renew group certificates.

IV. ANALYTICAL MODEL

In this section, we derive the probability of false positives and false negatives of the proposed k-anonymity mechanism, i.e., the probability of an honest user being regarded as malicious one and the probability of not detecting and actual attacker. For the sake of clarity, table I presents the specific notation considered from now on.

TABLE I
NOTATION

r	number of rounds
n	number of users in the system
a	number of malicious users
p	prob. of a malicious user performing an attack in a round r
k	number of users in a group
f	number of flags needed to revoke a user
t	time of the certificate expiration
p_h	prob. of an honest user being flagged in a round
p_a	prob. of an attacker being flagged in a round
$p_h^{f,r}$	prob. of a honest user being flagged f times after r rounds
$p_a^{f,r}$	prob. of an attacker being flagged f times after r rounds
FP	r.v. number of false positives in r rounds and f flags
FN	r.v. number of false negatives in r rounds and f flags

Given these definitions, we denote as p_h and p_a the probabilities of an honest user and an attacker, respectively, receiving a flag in a given round r, and can be computed as in (1) and (2) with $\alpha_h = \min(k-1, a)$ and $\alpha_a = \min(k-1, a-1)$.

$$\begin{aligned} p_h &= \\ &= \sum_{i=1}^{\alpha_h} \binom{k-1}{i} \prod_{j=1}^{i} \tfrac{a-j+1}{n-j} \prod_{j=1}^{k-i-1} \tfrac{n-a-j}{n-i-j} \left(1-(1-p)^i\right) \\ &= \sum_{i=1}^{\alpha_h} \binom{k-1}{i} \tfrac{a!\,(n-a-1)!\,(n-k)!}{(a-i)!\,(n-1)!\,(n-a-k+i)!} \left(1-(1-p)^i\right) \end{aligned} \tag{1}$$

$$\begin{aligned} p_a = p \cdot 1 + (1-p) \cdot \\ \sum_{i=1}^{\alpha_a} \binom{k-1}{i} \prod_{j=1}^{i} \tfrac{a-j}{n-j} \prod_{j=1}^{k-i-1} \tfrac{n-a-j+1}{n-i-j} \left(1-(1-p)^i\right) \end{aligned} \tag{2}$$

Then, we can compute the probability of an honest user and a real attacker being regarded as attackers after r rounds as in (3) and (4), respectively.

$$p_h^{f,r} = \sum_{i=f}^{r} \binom{r}{i} p_h^i (1-p_h)^{r-i} \tag{3}$$

$$p_a^{f,r} = \sum_{i=f}^{r} \binom{r}{i} p_a^i (1-p_a)^{r-i} \tag{4}$$

From the above equations, it can be easily derived the probabilities of false positives and false negatives as a function of the number of rounds r, i.e., the probability of a honest user being flagged as an attacker and the probability of an attacker being regarded as a honest user after r rounds.

The probability of false positive after r rounds p_{fp}^r is the probability of at least one honest user having more than f flags, which is equal to 1 minus the probability of all honest users having less than f flags, and can be expressed as in (5).

$$p_{fp}^r = 1 - \left(1 - p_h^{f,r}\right)^{(n-a)} \tag{5}$$

Analogously, the probability of false negative p_{fn}^r is the probability of at least one attacker having less than f flags after r rounds, which is equal to 1 minus the probability of all attackers having f or more flags, and can be expressed as in (6).

$$p_{fn}^r = 1 - \left(p_a^{f,r}\right)^a \tag{6}$$

In order to analyze with more precision the goodness of the mechanism, it can be useful to estimate the expected number of false positives and false negatives and their variance, as a function of the number of flags f and the number of rounds r. Let us define $FP_{f,r}$ and $FN_{f,r}$ as two discrete random variables following a binomial distribution that account for the values of false positives and false negatives respectively. Then, we can define their respective probability mass functions as in (7)), with expected values μ as in (8) and variance σ^2 as in (9).

$$\begin{aligned} f_{FP_{f,r}}(i) &= \mathrm{P}(FP_{f,r} = i) \\ &= \binom{n-a}{i} \cdot (p_h^{f,r})^i \cdot (1-p_h^{f,r})^{n-a-i} \\ f_{FN_{f,r}}(i) &= \mathrm{P}(FN_{f,r} = i) \\ &= \binom{a}{i} \cdot (1-p_a^{f,r})^i \cdot (p_a^{f,r})^{a-i} \end{aligned} \tag{7}$$

$$\begin{aligned} \mu_{FP} &= \mathrm{E}[FP_{f,r}] = \sum_{i=1}^{n-a} i \cdot \mathrm{P}(FP_{f,r} = i) \\ \mu_{FN} &= \mathrm{E}[FN_{f,r}] = \sum_{i=1}^{a} i \cdot \mathrm{P}(FN_{f,r} = i) \end{aligned} \tag{8}$$

$$\begin{aligned} \sigma^2_{FP_{f,r}} &= \mathrm{V}[FP_{f,r}] = \sum_{i=1}^{n-a} (i - \mu_{FP_{f,r}})^2 \cdot f_{FP_{f,r}}(i) \\ \sigma^2_{FN_{f,r}} &= \mathrm{V}[FN_{f,r}] = \sum_{i=1}^{a} (i - \mu_{FN_{f,r}})^2 \cdot f_{FN_{f,r}}(i) \end{aligned} \tag{9}$$

V. Performance evaluation

In this section we first validate in section V-A the analytical model presented in section IV via simulation, and then in section V-B we evaluate the goodness of our proposal when applied to a real scenario such as the current Spain's vehicle fleet.

In order to make the simulation easier, we have assumed in the following that an attacker always attacks in a round, that is to say that the protocol operation properly detects all the attackers and thus cannot lead to false negatives. However, there are still chances of leading to false positives (honest users flagged as attackers) and therefore the following analysis mainly focuses on the mean and variance of false positives.

A. Validation of the analytical model

Figure 1 shows the mean number of false positives obtained by simulation (100 iterations per each possible combination) in dashed line vs the mean and standard deviation in continuous line obtained by the analytical model. Due to space constraints, we present just a specific case; the purpose of it is just to show that the analytical model properly fits the protocol behavior.

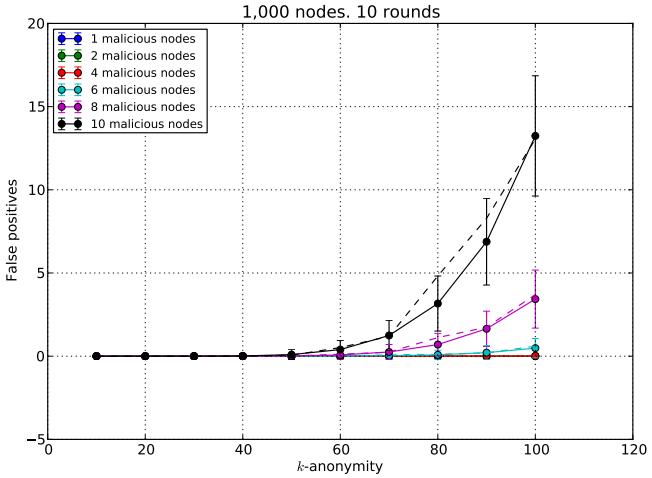


Fig. 1. False positives after 10 rounds with 1,000 nodes

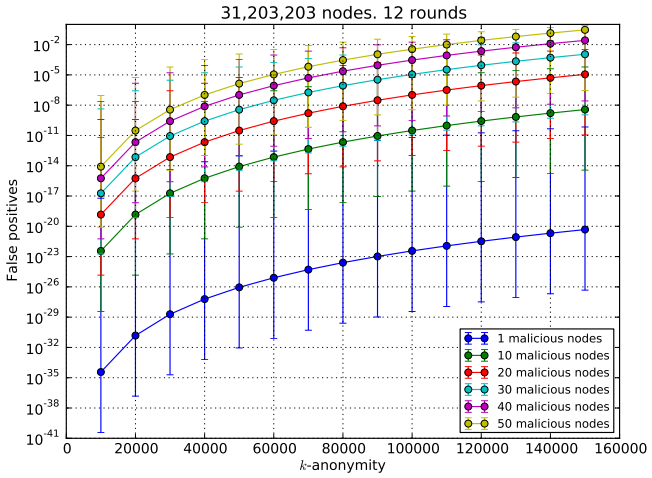


Fig. 2. Mean and STD of false positives vs k-anonymity after 12 rounds for a varying number of attackers

Obviously, it also fits in any other case but in the figures we present the average number of false positives after 10 rounds of operation in a network with 1000 nodes, a varying number of malicious nodes from 1 to 10 which are selected randomly and values k of anonymity ranging from 10 to 100 nodes per group also selected randomly. As per the figure, one can clearly notice that simulation values are within the range of expected values as per the analytical model.

Once showed the validity of the analytical model, in the following we evaluate the goodness of our proposal applied to the Spain's vehicle fleet.

B. Application to the Spain's car fleet

In this section we present the results obtained from the analytical model for the Spain's vehicle fleet in 2012[6], which rises up to 31,203,203 vehicles.

Figure 2 shows the mean and standard deviation of false positives obtained for a varying number of concurrent attackers

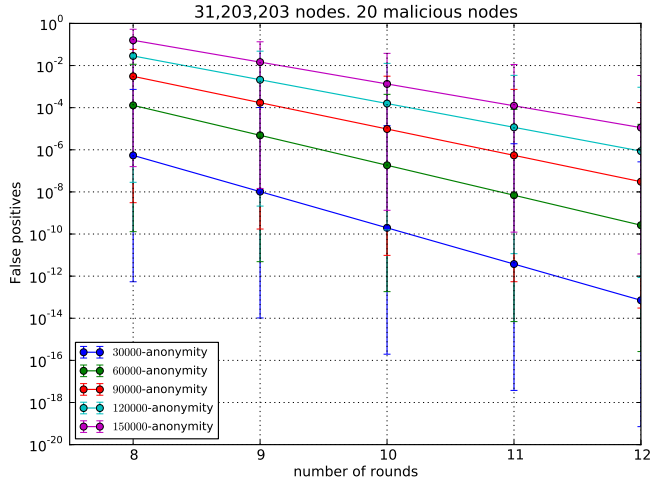


Fig. 3. Mean and STD of false positives vs rounds with 20 concurrent malicious nodes for a varying value of k-anonymity

ranging from 1 to 50 and k-anonymity from 10,000 to 150,000.

First impressions from the figure are that, as expected, the performance of the protocol increases with the number of nodes (now is 30,000 times greater than in section V-A). This is because, for the same level of k-anonymity, the number of groups increases, and therefore the probability of an honest user coinciding in every group with a malicious node diminishes.

Following the above reasoning and the results in the figure, the protocol performance decreases with the value of k-anonymity. That is to say that the more anonymity the worse performance. However, the average number of false positives is bounded to less than 1 (less than $3.205 \cdot 10^{-6}\%$) even for pretty high values of k-anonymity, which may satisfy most of the anonymity policies.

Figure 3 shows the evolution during time (rounds) of the number of false positives (mean and standard deviation) for 20 concurrent attackers and k-anonymity ranging from 30,000 to 150,000. The conclusion here is clear: the average number of false positives decreases in almost two orders of magnitude in every single round; and this is very promising result. Obviously, more rounds mean more time to detect attackers, but just a few rounds make negligible the probability of flagging an honest user as an attacker.

VI. Conclusions

In a VANET, every vehicle must own valid credentials issued by a trusted third party or CA in order to allow users to authenticate data. However, the use of credentials linked to vehicles may violate users' privacy as long as it facilitates the vehicle tracking. This is a risk that can be overcome by means of anonymity schemes.

The use of anonymity schemes can mitigate the risk of vehicles tracking; however, it comes at the expenses of: on the one hand, preventing VANET authorities (CA and TAs) from identifying malicious users and revoking them from the

network; or on the other hand, to discard the anonymity of users in front of the authorities thus to allow their revocation.

In this paper we have presented a method based on k-anonymity that preserves the vehicles' anonymity both against other vehicles/users of the system and the authorities while still being able to track malicious users and revoke them.

For the evaluation of the proposal, we have derived an analytical model for the number of false positives and negatives in several scenario conditions, and we have validated the model by simulation. Then we have analyzed the performance of our proposal with a real country vehicle fleet (the Spanish one) leading to quite promising results in terms of malicious tracking efficiency while providing good levels of k-anonymity. Provided method can effectively identify malicious users whenever they misbehave a given number of times with almost negligible rates of false positives.

Other directions for future work include the development and evaluation of possible attacks to the system, in parallel with an investigation of more efficient and secure schemes.

ACKNOWLEDGMENT

This work was partially supported by the Spanish *Ministerio de Economía y Competitividad*, the Spanish *Comisión Interministerial de Ciencia y Tecnología*, the Spanish *Ministerio de Industria, Energía y Turismo*, the *Generalitat de Catalunya* and European FEDER funds under the projects: MINECO TUERI (TIN2011-25452), CICYT TAMESIS (TEC2011-22746), INNPACTO DEPHISIT (IPT-2012-0585-370000), CONSOLIDER ARES (CSD2007-00004), as well as the grant 2009 SGR-1362 to consolidated research groups, the funding of which is gratefully acknowledged.

REFERENCES

[1] TomTom user data sold to Dutch police, used to determine ideal locations for speed traps. http://www.engadget.com/2011/04/27/tomtom-user-data-sold-to-danish-police-used-to-determine-ideal/.

[2] P. Caballero-Gil, C. Caballero-Gil, and J. Molina-Gil. How to build vehicular ad-hoc networks on smartphones. *Journal of Systems Architecture*, 59(10, Part B):996 – 1004, 2013.

[3] E. C. CRYPTOGRAPHY. The advantages of elliptic curve cryptography for wireless security. *IEEE Wireless Communications*, page 63, 2004.

[4] V. Daza, J. Domingo-Ferrer, F. Sebe, and A. Viejo. Trustworthy privacy-preserving car-generated announcements in vehicular ad hoc networks. *Vehicular Technology, IEEE Transactions on*, 58(4):1876–1886, May 2009.

[5] M. de Fomento Gobierno de España. Encuesta de movilidad de las personas residentes en España (movilia 2006/2007). http://www.fomento.es/MFOM/LANG_CASTELLANO/ESTADISTICAS_Y_PUBLICACIONES/INFORMACION_ESTADISTICA/Movilidad/Movilia2006_2007/default.htm.

[6] D. G. de Tráfico (DGT). Parque de vehículos por ccaa, provincias y tipos, 2012.

[7] Google. Google Maps, traffic option. https://support.google.com/maps/answer/ 61454?hl=en.

[8] D. Huang, S. Misra, M. Verma, and G. Xue. Pacp: An efficient pseudonymous authentication-based conditional privacy protocol for vanets. *Intelligent Transportation Systems, IEEE Transactions on*, 12(3):736–746, Sept 2011.

[9] R. Lu, X. Lin, H. Zhu, P.-H. Ho, and X. Shen. Ecpp: Efficient conditional privacy preservation protocol for secure vehicular communications. In *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, pages –, April 2008.

[10] K. Oishi, M. Mambo, and E. Okamoto. Anonymous public key certificates and their applications. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 81(1):56–64, 1998.

[11] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J.-P. Hubaux. Eviction of misbehaving and faulty nodes in vehicular networks. *Selected Areas in Communications, IEEE Journal on*, 25(8):1557–1568, Oct 2007.

[12] F. Schaub, F. Kargl, Z. Ma, and M. Weber. V-tokens for conditional pseudonymity in vanets. In *IEEE Wireless Communications and Networking Conference (WCNC 2010)*, pages 1–6, Los Alamitos, April 2010. IEEE Computer Society Press.

[13] F. Sebé-Feixas. Privacy in vehicular networks and location-based services. URV Chairs - Summer Courses, June 2007.

[14] L. Sweeney. k-anonymity: a model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.*, 10(5):557–570, Oct. 2002.

[15] Sygic. Traffic service. http://www.sygic.com/en/android:traffic.

[16] TomTom. HD traffic. http://www.tomtom.com/en_us/services/live/hd-traffic/.

[17] A. Wasef and X. Shen. Edr: Efficient decentralized revocation protocol for vehicular ad hoc networks. *Vehicular Technology, IEEE Transactions on*, 58(9):5214–5224, Nov 2009.

[18] WAZE. Real-Time maps and traffic information based on the wisdom of the crowd. http://www.waze.com/.

[19] S. Zakhary, M. Radenkovic, and A. Benslimane. The quest for location-privacy in opportunistic mobile social networks. In *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*, pages 667–673, 2013.

[20] L. Zhang, Q. Wu, A. Solanas, and J. Domingo-Ferrer. A scalable robust authentication protocol for secure vehicular communications. *Vehicular Technology, IEEE Transactions on*, 59(4):1606–1617, May 2010.

Sistema de telepeaje en zonas urbanas

Roger Jardí-Cedó*, Macià Mut-Puigserver†, M. Magdalena Payeras-Capellà†, Jordi Castellà-Roca*, Alexandre Viejo*

* Dpt. d'Enginyeria Informàtica i Matemàtiques, UNESCO Chair in Data Privacy,
Universitat Rovira i Virgili,
Av. Països Catalans 26,E-43007 Tarragona, Spain
Email: {roger.jardi,jordi.castella,alexandre.viejo}@urv.cat

† Dpt. de Ciències Matemàtiques i Informàtica,
Universitat de les Illes Balears,
Ctra. de Valldemossa, km 7,5. E-07122 Palma de Mallorca, Spain
Email: {macia.mut, mpayeras}@uib.es

***Resumen*—Las Low Emission Zones (LEZ) limitan el acceso de vehículos a las zonas más céntricas de las ciudades con el objetivo de reducir la densidad del tráfico y la contaminación ambiental. Estos sistemas tienen problemas de privacidad de los conductores y de efectividad en la detección del fraude. Este artículo presenta un sistema de telepeaje para LEZ que mejora estos problemas.**

***Palabras clave*—Low Emission Zones, Privacidad, Seguridad, Telepeaje**

I. Introducción

Ciudades como París, Barcelona o Roma tienen problemas de circulación, con grandes atascos, y problemas de contaminación debidos a la gran concentración de vehículos en ciertas zonas. Las directrices sobre calidad del aire elaboradas por la OMS en 2005 orientan "sobre la manera de reducir los efectos de la contaminación del aire en la salud". Basadas en estas recomendaciones, diferentes directivas europeas, como la 2008/50/CE, limitan el nivel de ciertos contaminantes ambientales. Para cumplir esta legislación, las diferentes administraciones están implantando, entre otras medidas, carriles de alta ocupación [1], velocidad variable o restricciones de circulación en zonas céntricas. Esta última medida, conocida como **Low-Emission Zone (*LEZ*)** y adoptada en varias ciudades[1], establece que los vehículos paguen por circular en función de ciertas condiciones, como su peso o emisiones.

Desde hace décadas, el telepeaje electrónico o Electronic Toll Collenction (ETC) ha sido utilizado en autopistas, túneles o puentes para agilizar el pago en los peajes, y a su vez, reducir los atascos. Por otro lado, gracias a nuevas tecnologías como el GPS y la comunicación inalámbrica, se han desarrollado los vehicular location-based services (VLBS), que tienen el propósito de proporcionar información a los conductores en función de su ubicación geográfica y mejorar la eficiencia del transporte. Los sistemas ETC, entendidos como VLBS, son conocidos como **Electronic Road Pricing (ERP)** y presentan varias mejoras como un cálculo más flexible de las tasas dependiendo de la distancia recorrida, la ruta o el tiempo. Además, estos sistemas, aplicados a zonas urbanas, permiten la gestión del tráfico en zonas céntricas mediante el control del flujo y la densidad de los vehículos, reduciendo así atascos. Esto se consigue modificando el precio de las tasas de forma dinámica (el aumento del precio de las zonas más densas sugiere a los conductores evitarlas). No obstante, como se verá, estos sistemas tienen ciertos problemas de privacidad.

[1] http://lowemissionzones.eu/

I-A. Estado del arte

En los últimos años se han propuestos, en la literatura, varios sistemas *ERP* ([2], [3], [4], [5], [6], [7]). Todos ellos requieren el uso de una On-Board Unit (OBU) con GPS y un sistema de comunicación inalámbrica con el fin de recoger y enviar al proveedor del servicio *SP* (ver definición en II-A) información relacionada con la localización del vehículo y las tasas a pagar. Es decir, la tarificación es en función de la ruta del vehículo. En [2] y [3], la *OBU* envía información del camino recorrido a un servidor externo, propiedad del *SP*, el cual tarifica de acuerdo a su trayectoria en cada periodo de facturación. En [4], [5], [6], [7], las tasas son calculadas localmente en cada *OBU* y son enviadas al servidor del *SP* en cada periodo de tarificación. En este caso, la revelación de información relacionada con la localización del vehículo es mínima. Para conseguirlo, se basan en el uso de pruebas criptográficas para demostrar que la *OBU* ha sido honesta en el cálculo y la agregación de las tasas.

El control del fraude es un objetivo importante de los sistemas *ERP*. Los conductores, para ahorrar dinero, podrían actuar de forma malintencionada (p.ej. desconectando o modificando datos de la *OBU*). Por este motivo, se implementan mecanismos basados en puntos de control *Chp*s con la intención de poner a prueba su honestedad. Los *Chp*s, situados aleatoriamente en la carretera y equipados con cámaras, registran las matrículas de todos los vehículos que los atraviesan. Estas fotos son pruebas que sitúan a un vehículo en un determinado momento y lugar, y son utilizadas para verificar que la trayectoria de un vehículo no ha sido alterada. Para ello, en

el periodo de facturación el *SP* y conductor interaccionan. La detección del fraude tiene una cierta probabilidad y depende de la cantidad de *Chp*s. Además, el desconocimiento del número y de su localización, por parte de los conductores, es básico.

El nivel de privacidad de los conductores y de detección del fraude son un compromiso. Si se desea un grado de detección elevado, la privacidad se ve afectada. Es decir, el *SP* será capaz de reconstruir la trayectoria de un vehículo con más precisión si la cantidad de *Chp*s es mayor. Además, si los *Chp*s son movidos aleatóriamente cada cierto tiempo y los trayectos de los vehículos siguen una rutina (p.ej. ir al trabajo), la precisión podría ser aún mayor pero la privacidad se vería afectada.

I-B. Contribuciones y estructura del documento

Este artículo propone un sistema *ERP* para *LEZ*s con el objetivo de mejorar el control del fraude y la privacidad de los conductores honestos mediante anonimidad revocable. A diferencia de los otros sistemas, los *Chp*s, equipados con cámaras, registran únicamente los vehículos fraudulentos, manteniendo así la privacidad de los conductores honestos. Además, la *OBU* del vehículo no registra su ubicación, no se requiere de reconciliación entre conductor y sistema en la fase de facturación, y el control del fraude es no probabilístico.

Estructura: En la Sec. II se presenta el sistema. En la Sec. III se introduce el protocolo. En la Sec. IV se evalúa la seguridad, y en la Sec. V se presenta las conclusiones.

II. Modelo del sistema

II-A. Participantes involucrados

- *Conductor D:* Conduce un vehículo por una *LEZ*.
- *Vehículo V:* Está registrado con un único *D* aunque puede ser conducido por varios *D*s. *V* tiene un identificador (la matrícula) que lo enlaza con el propietario.
- *Secure element SE*: Módulo de seguridad a prueba de manipulaciones, instalado en cada *V* por la autoridad de tráfico competente. Realiza operaciones sensibles para garantizar los requisitos de seguridad del sistema.
- *On-board unit OBU:* Dispositivo de capacidad de computación y almacenamiento superior al *SE* instalado en cada *V*. Conecta el *SE* con el exterior y realiza las operaciones menos sensibles del protocolo. Dispone de un módulo de localización (GPS).
- *Service Provider SP:* Ofrece un servicio de cobro electrónico de peajes (*ERP*) para zonas urbanas gracias a una concesión pública de la administración local (p.ej. un ayuntamiento). Esta entidad, a parte de tener el derecho de ofrecer este servicio, tiene la responsabilidad de gestionar el sistema.
- *Checkpoint Chp:* Está instalado en la zona restringida por el *SP* y tiene como objetivo controlar el acceso de los *V*s que entran/salen de la zona para evitar el fraude.
- *Vehicle certification authority VCA:* Proporciona claves y certificados a los *Vs*.
- *Punisher authority PA:* Entidad de confianza que conoce la identidad del propietario del *V* y la revela si hay fraude.

II-B. Requisitos

A continuación se describen los requisitos del sistema, relacionados con el fraude, la privacidad, la autenticidad y la tecnología, para establecer las bases del sistema.

II-B1. Requisitos anti-fraude: Cuando un *V* entra o sale de una *LEZ* a través de un *Chp*, ambos obtienen una **prueba de entrada** γ_i o **prueba de salida** γ_o. Esta γ_i es información que demuestra que un determinado *V* entró a la *LEZ* por un determinado *Chp* a una determinada hora. Esta *prueba* se considera **válida** si no puede ser modificada o alterada sin ser detectado una vez generada (íntegra), si sus emisores pueden demostrar que es suya (auténtica) y tampoco pueden negar su autoría (no-repudiable). Cada prueba esta **vinculada** a un *V* y un *Chp*. La **vinculación** de una prueba con un *V* garantiza que el token no puede ser usado por otro *V'* de forma voluntaria o involuntaria. Esto evita la *duplicidad* de una misma prueba cuando es utilizada por más de un *V* al mismo tiempo.

El *SP* trabaja para asegurar que todos los *D*s paguen correctamente. En caso de no ser así, el *SP* identifica a los *D*s infractores y genera evidencias que lo demuestran. El **fraude** es cometido por un *D* cuando éste conduce por una *LEZ* sin ninguna γ_i, con una γ_i no válida, con una γ_i válida de otro *V*, o si en la salida no realiza el pago correctamente. Un *SP* tampoco puede **acusar falsamente** de fraude a un *D* honesto (*D* no debe estar indefenso). Una falsa acusación ocurre cuando un *SP* afirma injustamente que un *V* no tiene una γ_i, que tiene una prueba no válida, una prueba válida que pertenece a otro *V*, o si en la salida no realiza el pago correctamente.

II-B2. Requisitos de autenticidad: En la entrada y la salida de la *LEZ*, *V*s y *Chp*s intercambian información. Al establecer la comunicación, ambas partes, tanto *V* como *Chp*, deben demostrar su identidad a la otra parte. De esta forma, cada una puede estar segura de que la comunicación se realiza con la entidad que dice ser. En caso de no ser así, se deben tomar medidas para denunciar este hecho.

II-B3. Requisitos de privacidad: El control del fraude por parte del *SP* puede llegar a comprometer la privacidad de los *D*s. En este caso, la curiosidad del *SP* puede causar una excesiva monitorización del sistema e incluso ser consciente de cada recorrido que hace un *V*. Con el fin de evitar un excesivo control por parte del *SP* sobre los *V*s, el sistema debe (i) garantizar la privacidad (la identidad de *D* o *V* no puede ser enlazada con ningún recorrido de ningún *V*); (ii) evitar la trazabilidad (*SP* no debe conozer el recorrido de un *V*); y (iii) proveer al *D* de un anonimato revocable (si un *D* realiza fraude, el *SP* necesita su identidad para poder sancionarlo, sólo entonces, es revelada).

II-B4. Requisitos funcionales: La *tecnología para comunicar V*s y *Chp*s entre ellos debe permitir, a estos últimos, comunicarse con el *V* más cercano a ellos. Esto se podría conseguir combinando tecnologías de comunicación de corta/media distancia, tales cómo Wimax, ZigBee IEEE 802.15.4 o Bluetooth IEEE 802.15.1, con el uso de antenas direccionales o por triangulación, por ejemplo. La *comunicación* y el *cálculo* requerido en el protocolo deberán ser lo

suficientemente rápidos para permitir una intercomunicación en movimiento entre *V*s y *Chp*s. Cualquier *interacción* con el *D* deberá ser ágil y fácil. El *sistema de pago electrónico* requerido en el sistema deberá ser anónimo y no trazeable. Además, deberá ser lo suficientemente rápido para dar tiempo a realizar la transacción en el proceso de salida de la zona más externa de la *LEZ*.

II-C. Modelo de los adversarios

Los intereses de *D*s y *SP* pueden ser opuestos. Por un lado, los *D*s quieren ahorrar dinero, a veces de forma deshonesta y actuando en contra del sistema. Por otro lado, el *SP* puede llegar a comprometer la privacidad de los *D*s, ya que conocer la identidad de los *V*s, en caso de fraude, le puede ser útil. Además, el *SP*, con el deseo de ganar más dinero, puede actuar deshonestamente contra los *V* acusandolos de fraude de manera infundada. Por consiguient, el control del fraude y la protección de la privacidad pueden llegar a ser objetivos opuestos.

III. Descripción del protocolo

Antes de iniciar el sistema, las entidades del sistema son inicializadas (III-A: Setup y III-B: Certificación). Además, el *SP* fija los precios de la *LEZ* (III-D: Generación de precios), por unidad de tiempo y categoría de emisiones, enviando una lista de precios, firmada por la entidad competente, a cada *Chp*. El *SP*, cada vez que decida actualizar los precios, repetirá estas mismas operaciones.

El *SE* genera unas credenciales diferentes para *V* cada vez que entra a una *LEZ* (III-C: Generación de los certificados) para poder autenticarse correctamente con los *Chp*s que gestionan las entradas y salidas de la *LEZ*.

Cuando un vehículo *V* entra a una *LEZ* (III-E: Entrada al sistema) se comunica con un *Chp* y se autentican mutuamente. Si la autenticación con *V* falla, únicamente en esta situación, el *Chp* toma una fotografía de la matrícula del *V* como evidencia de la infracción y con ella, genera una prueba de incidencia de entrada ζ_i. La ζ_i es enviada al *PA* para verificar la existencia de fraude y proceder con la sanción. Si la autenticación es correcta, el *V* obtiene una prueba de entrada γ_i que contiene el tiempo de entrada.

Cuando un *V* sale de la *LEZ* (III-F: Salida del sistema) se comunica con un *Chp* y se autentican mutuamente. Si la autenticación con *V* es correcta, el *Chp* informa al *V* del tiempo de salida y de la cuenta destino para realizar el pago. Con dicha información, *V* calcula el importe a pagar por el tiempo de estancia y categoría de emisiones, y realiza una transacción mediante un sistema de pago electrónico. La referencia de la transacción es enviada al *Chp* como prueba del pago. Finalmente, *V* recibe una prueba de salida γ_o como recibo. Si la autenticación falla, el *Chp* toma una fotografía de *V* que forma parte de una prueba de incidencia de salida ζ_o, y la envía a *PA*.

La verificación del pago se realiza por el *SP* (III-G: Verificación del pago) a posteriori y cada cierto tiempo. Por cada pareja γ_i y γ_o asociada a un mismo *V*, el *SP* comprueba si el valor de la transacción coincide con la tarificación dependiendo del tiempo de estancia y la categoría de emisiones. Si no es correcto, genera una prueba de incidencia de pago ζ_p con estos registros y la envía a *PA*.

Cuando *PA* recibe una ζ (III-H: Sanción), la verifica. Si hay fraude, *PA* revela la identidad del propietario del *V* (revoca el anonimato) y le solicita pruebas que desmientan la acusación por parte de *SP*. En función de éstas, *PA* sanciona o no al propietario.

III-A. Setup

El proceso de setup es el siguiente:

1. *PA* obtiene de las autoridades competentes (p.ej. Policía):
   - Una pareja de claves asimétricas (Pk_{PA},Sk_{PA}) y un certificado de clave pública $cert_{PA}$
   - Un repositorio de certificados de las autoridades.
2. *SP* y *VCA* obtienen de las autoridades competentes (p.ej. Ayuntamiento y DGT):
   - Una pareja de claves (Pk_{SP},Sk_{SP}) y (Pk_{VCA},Sk_{VCA}), y un certificado de CA ($cert_{SP}$ y $cert_{VCA}$) emitido por las autoridades.
   - Un repositorio de certificados de las autoridades.

   La longitud de la cadena de certificación de *VCA* es 1, y 0 en el caso de *SP*. La duración de $cert_{SP}$ puede coincidir con el tiempo de concesión del servicio, sin excederlo.
3. *VCA*:
   I. Define un conjunto de vehículos $V = \{v_1, v_2, ..., v_{n_V}\}$, donde $n_V = |V|$ es la cantidad de vehículos.
   II. Define una colección de subconjuntos $K = \{C_1, C_2, ..., C_{n_K}\}$ partición de *V*, dónde $n_K = |K|$, con $|C_i| = n_C$, $\forall i$
   III. Genera y asocia una entidad de certificación VCA_{C_i} a cada elemento de la colección *K* ($C_1, ..., C_{n_K}$):
      iii.I. Una pareja de claves ($Pk_{VCA_{C_i}}$, $Sk_{VCA_{C_i}}$), $\forall i \in \{1, ..., n_K\}$
      iii.II. Un certificado de CA $cert_{VCA_{C_i}}$, $\forall i \in \{1, ..., n_K\}$, con una duración c_{exp} y una longitud de la cadena de certificación de 0.
4. Cada *Chp* realiza las siguientes operaciones:
   I. Obtiene un repositorio de certificados de las autoridades y entidades (excepto de los vehículos).
   II. Genera una pareja de claves (Pk_{Chp},Sk_{Chp})
   III. Obtiene de *SP* un certificado de clave pública $cert_{Chp}$, conteniendo una extensión $cert_{Chp}.loc$ con sus coordenadas de localización, de manera segura.

III-B. Certificación

Se asume que el *SE* de cada *V* ha sido inicializado previamente con un repositorio de certificados de las autoridades de certificación, con un identificador del vehículo V_{id} y con sus especificaciones técnicas (marca, modelo, número de chasis, matricula, emisiones de CO_2 y gases contaminantes, etc.).

El proceso de certificación de un *V* es realizado por *VCA*, al comprar el vehículo y/o al pasar la Inspección Técnica de Vehículos (ITV):

1. Registra el *V* a un elemento del subconjunto *K* (a un C_i)
2. Descarga, en el *SE* del *V*, la entidad de certificación VCA_{C_i} asociada al C_i (consistente en $Pk_{VCA_{C_i}}$, $Sk_{VCA_{C_i}}$ y $cert_{VCA_{C_i}}$), mediante un canal de comunicación seguro.

III-C. Generación de certificados

Esta fase se realiza cada vez antes de que un *V* entre a una *LEZ*. Gracias a la entidad de certificación VCA_{C_i} instalada en el *SE* del *V* en la fase anterior, éste les permite realizar las siguientes operaciones para generar nuevos certificados de clave pública:

1. Calcula una nueva pareja de claves (Pk_{V_q}, Sk_{V_q})
2. Genera un certificado de clave pública $cert_{V_q}$ con las siguientes características:
   - Con una extensión $cert_{V_q}.idS$ que contiene el cifrado probabilístico (p.ej. usando OAEP padding [8], estandarizado en PKCS #1v2 y RFC 2437) del identificador del vehículo V_{id} con la clave pública de *PA*: $Enc_{Pk_{VCA}}(V_{id})$
   - Con una extensión $cert_{V_q}.emis$ que contiene la categoría de emisiones de CO_2 del vehículo.

III-D. Generación de precios

Cada vez que el *SP* modifica los precios de tarificación de una *LEZ* realiza los siguientes pasos:

1. Fija los *precios* por unidad de tiempo y categoría de emisiones (p.ej. european emission standards).
2. Genera un timestamp *ts*
3. Compone una información de precios $\theta = (precios, ts)$
4. Firma el θ: $Sign_{SP}(\theta) = \bar{\theta}$
5. Envía θ^*=(θ, $\bar{\theta}$) a cada *Chp*

III-E. Entrada al sistema

Cuando un *Chp* situado en la entrada de una *LEZ* detecta un *V*, se inicia el siguiente protocolo:

1. *Chp*:
   I. Genera un nonce N_A
   II. Compone una información de entrada $\psi = (N_A, \theta^*)$
   III. Firma el ψ: $Sign_{Chp}(\psi) = \bar{\psi}$
   IV. Envía ψ, $\bar{\psi}$ y su $cert_{Chp}$ a *V*
2. El *SE* del *V* con la ayuda de su *OBU*:
   I. Verifica el certificado $cert_{Chp}$ y la firma $\bar{\psi}$: $Verif_{Chp}(N_A,\theta^*,\bar{\psi})$
   II. Verifica la firma $\bar{\theta}$: $Verif_{SP}(precios,ts,\bar{\theta})$
   III. Verifica las coordenadas de localización $cert_{Chp}.loc$ del *Chp* (incluido en su certificado).
   IV. Genera un nonce N_B y calcula el fingerprint $fing_{Chp}$ de $cert_{Chp}$ (se calcula con la función hash del certificado y sirve de identificador).
   V. Compone un mensaje $\omega_{V_q} = (\theta^*, N_A, N_B, fing_{Chp})$
   VI. Firma ω_{V_q}: $Sign_{V_q}(\omega_{V_q}) = \omega_{\bar{V}_q}$
   VII. Envía N_B, $\omega_{\bar{V}_q}$ y $cert_{V_q}$ a *Chp*
3. *Chp*:
   I. Genera un timestamp ts'
   II. Verifica el certificado $cert_{V_q}$ y la firma $\omega_{\bar{V}_q}$: $Verif_{V_q}(\theta^*, N_A, N_B, fing_{Chp}, \omega_{\bar{V}_q})$
4. Si alguna de las verificaciones falla, el *Chp* realiza:
   I. Genera un numero de incidencia de entrada in_i
   II. Toma una fotografía *ph* de la matrícula de *V*
   III. Procesa la matricula mat
   IV. Compone una prueba de incidencia de entrada ζ_i= (in_o,mat,ph,ts',θ^*,N_A,N_B,$fing_{Chp}$,$\omega_{\bar{V}_q}$,$cert_{V_q}$)
   V. Firma ζ_i: $Sign_{Chp}(\zeta_i) = \bar{\zeta}_i$
   VI. Envía $\zeta_i{}^*$=(ζ_i, $\bar{\zeta}_i$) y su $cert_{Chp}$ a *SP*
5. Si las verificaciones realizadas en 3) son correctas entonces,
   I. El *Chp*:
      i.I Calcula el $fing_{V_q}$ de $cert_{V_q}$
      i.II Compone una prueba de entrada γ_i = (θ^*,N_A,N_B,$fing_{Chp}$,$\omega_{\bar{V}_q}$,$fing_{V_q}$,ts')
      i.III Firma γ_i: $Sign_{Chp}(\gamma_i) = \bar{\gamma}_i$
      i.IV Envía ts' y $\bar{\gamma}_i$ a *V*
   II. El *SE* del *V* con la ayuda de la *OBU*:
      ii.I Verifica la firma $\bar{\gamma}_i$: $Verif_{Chp}(\theta^*, N_A, N_B, fing_{Chp}, \omega_{\bar{V}_q}, fing_{V_q}, ts', \bar{\gamma}_i)$
      ii.II Verifica ts' sea reciente:$|ts'$-*current time*$| < \delta$, donde δ es tiempo fijado
      ii.III Guarda γ_i^* = (γ_i,$\bar{\gamma}_i$)

III-F. Salida del sistema

Cuando un *Chp* situado en la salida de una *LEZ* detecta un *V*, se inicia el siguiente protocolo:

1. *Chp*:
   I. Genera un timestamp ts'' y un nonce N_C
   II. Compone un información de pago ρ = (ts'',N_C,acc), donde acc identifica la cuenta destino, del sistema de pago electrónico asumido, de *SP*
   III. Firma el ρ: $Sign_{Chp}(\rho) = \bar{\rho}$
   IV. Envía ρ, $\bar{\rho}$ y su $cert_{Chp}$ a *V*
2. El *SE* del *V* con la ayuda de su *OBU*:
   I. Verifica el certificado $cert_{Chp}$ y la firma $\bar{\rho}$: $Verif_{Chp}(ts'', N_C, acc, \bar{\rho})$
   II. Verifica las coordenadas de localización $cert_{Chp}.loc$ del *Chp* (incluido en su certificado).
   III. Verifica ts'' sea reciente: $|ts''-$*current time*$| < \delta$
   IV. Recupera el ts' del último registro γ_i
   V. Calcula el tiempo de estancia τ a LEZ: (ts''-ts')=τ
   VI. Recupera los *precios* contenido en el θ^* del γ_i
   VII. Calcula y acumula en *amount* la cantidad de dinero a pagar en función de τ, de sus emisiones y los *precios*
   VIII. Realiza una transferencia según *amount* a la cuenta destino acc y obtiene una referencia *trans*
   IX. Genera un nonce N_D y calcula el $fing_{Chp}$

X. Compone un mensaje ω_{V_q} = (ts'', N_C, N_D, $fing_{Chp}$,*trans*)
XI. Firma ω_{V_q}: $Sign_{V_q}(\omega_{V_q}) = \bar{\omega_{V_q}}$
XII. Envía N_D, *trans*, $\bar{\omega_{V_q}}$ y $cert_{V_q}$ a *Chp*

3. *Chp* verifica el certificado $cert_{V_q}$ y la firma $\bar{\omega_{V_q}}$: $Verif_{V_q}(ts'', N_C, N_D, fing_{Chp}, trans, \bar{\omega_{V_q}})$
4. Si alguna de las verificaciones falla, el *Chp* realiza:
   I. Genera un numero de incidencia de salida in_o
   II. Toma una fotografía *ph* de la matrícula de *V*
   III. Procesa la matrícula mat
   IV. Compone una prueba de incidencia de salida ζ_o=(in_o,mat,ph,ts'',N_C,N_D,$fing_{Chp}$,$trans$, $\bar{\omega_{V_q}}$,$cert_{V_q}$)
   V. Firma ζ_o: $Sign_{Chp}(\zeta_o) = \bar{\zeta_o}$
   VI. Envía $\zeta_o{}^*$=(ζ_o, $\bar{\zeta_o}$) y su $cert_{Chp}$ a *SP*
5. Si las verificaciones realizadas en 3) son correctas entonces,
   I. El *Chp*:
      i.I Calcula el $fing_{V_q}$ de $cert_{V_q}$
      i.II Compone una prueba de salida γ_o = (ts'', N_C,N_D,$fing_{Chp}$,$trans$,$\bar{\omega_{V_q}}$,$fing_{V_q}$)
      i.III Firma γ_o,$Sign_{Chp}(\gamma_o)$=$\bar{\gamma_o}$, y lo envía a *V*
   II. El *SE* del *V*, con la ayuda de la *OBU*:
      ii.I Verifica la firma $\bar{\gamma_o}$: $Verif_{Chp}(ts''$,N_C,N_D, $fing_{Chp}$,$trans$,$\bar{\omega_{V_q}}$, $fing_{V_q}$,$\bar{\gamma_o}$)
      ii.II Guarda γ_o^* = (γ_o,$\bar{\gamma_o}$)

III-G. Verificación del pago

Cada *Chp* envía periódicamente todas las γ_i, γ_o y ζs (ζ_i y ζ_o) a *SP*. *SP* reenvia ζ_i y ζ_o a *PA*. Además, el *SP* realiza las siguientes verificaciones cada cierto tiempo (en batch) por cada conjunto de registros γ_i y γ_o asociados a un mismo V_q (con un mismo $fing_{V_q}$):

1. Extrae ts', ts'', y $cert_{V_q}.emis$, contenidos en γ_i y γ_o
2. Extrae *precios*, contenido en el θ^* del γ_i
3. Extrae la referencia *trans*, contenida en γ_o
4. Calcula el tiempo en la zona τ=ts''-ts'
5. Calcula la cantidad total a pagar $amount'$ en función de τ, $cert_{V_q}.emis$ y *precios*
6. Verifica si $amount = amount'$
7. Verifica que la transferencia se haya realizado
8. Verifica que *trans* haya sido utilizado en otra γ_o (p.ej. buscando duplicados)
9. Si la verificación falla,
   I. Genera un numero de incidencia de verificación in_v
   II. Compone una prueba de incidencia de pago ζ_p con γ_i y γ_o de V_q: ζ_p=(in_v,γ_i,γ_o). En caso que *trans* haya sido reutilizado, añade la γ_o' que lo demuestra ζ_p = (in_v, γ_i, γ_o, γ_o')
   III. Firma ζ_p: $Sign_{SP}(\zeta_p) = \bar{\zeta_p}$
   IV. Envía $\zeta_p{}^*$=(ζ_p,$\bar{\zeta_p}$) y su $cert_{SP}$ a *PA*

III-H. Sanción

PA realiza las siguientes operaciones en función del tipo de incidencia ζ recibida:

1. Si es una ζ_i o ζ_o:
   I. Verifica las firmas
   II. Recupera la matricula mat de la foto adjuntada
2. Si es una ζ_p:
   I. Verifica todas las firmas contenidas en ζ_p y que el firmante de γ_i y γ_o sea el mismo vehículo.
   II. Verifica el pago repitiendo los pasos 1-7 de III-G
   III. Si se trata de un pago duplicado, se verifica que γ_o y γ_o' contengan el mismo *trans*
   IV. Si ratifica la incidencia, recupera el identificador V_{id} de V_q (en caso de pago duplicado, se realiza sobre el último *V* en salir de *LEZ*) abriendo la extensión $certV_q.idS$ (del certificado $certV_q$ contenido en el registro γ_i o γ_o): $Dec_{PA}(certV_q.idS) = V_{id}$
3. Se pone en contacto con el propietario del *V* a partir de la mat o del V_{id}, le informa del proceso sancionador y le exige pruebas que lo refuten.
4. Si el propietario presenta pruebas, se evalúan. En caso de haber fraude, se le multa en función del tipo de infracción.

IV. Análisis de seguridad/requisitos

En esta sección analizamos las propiedades de seguridad de nuestro protocolo. La discusión esta organizada en tres proposiciones que con sus respectivas reivindicaciones proporcionan evidencias del cumplimento de las propiedades de seguridad del esquema.

Proposición 1. *La propuesta preserva la autenticidad, el no repudio y la integridad de las pruebas de entrada y de salida.*

Reivindicación 1. *No es posible la creación de pruebas de entrada o salida fraudulentas.*

Prueba. Las pruebas de entrada tienen la forma siguiente γ_i = (θ^*,N_A,N_B,$fing_{Chp}$,ω_{V_q},$fing_{V_q}$,ts'). El checkpoint firma la prueba de entrada γ_i: $Sign_{Chp}(\gamma_i) = \bar{\gamma_i}$ y envía el par ts' y $\bar{\gamma_i}$ al vehículo. Análogamente, la prueba de salida γ_o = (ts'', N_C,N_D,$fing_{Chp}$,$trans$,$\bar{\omega_{V_q}}$,$fing_{V_q}$) es firmada por el checkpoint, $Sign_{Chp}(\gamma_o)$=$\bar{\gamma_o}$,y enviada al vehículo. Por tanto la creación de pruebas de entrada y de salida falsas es computacionalmente imposible actualmente si no se dispone de la clave secreta utilizada por el *Chp* en la firma.

Reivindicación 2. *Los* Chp*s emisores de las pruebas de entrada y de salida no pueden negar las emisiones de las mismas.*

Prueba. Las pruebas de entrada son generadas y firmadas por su emisor (los *Chp*s) y, considerando que el esquema de firma es seguro, esta operación solamente la pueden realizar ellos. Por lo tanto, la identidad del Chp está asociada a la prueba de entrada o de salida y por las propiedades del esquema de firma electrónica estos no pueden negar su autoría.

Reivindicación 3. *El contenido de las pruebas de entrada y de salida no puede ser modificado por los vehículos.*

Prueba. Suponiendo que el esquema de firma es seguro y que la función resumen utilizada en la firma es resistente

a colisiones, si se modifica el contenido del billete la verificación de la firma de los billetes será incorrecta dado que $Sign_e(m) = E_{Sk_e}(h(m)) = \bar{m}$. Para que la verificación fuera correcta se debería volver a generar la firma realizada sobre el resumen del nuevo contenido. Esta operación no es posible so no se dispone de la clave secreta del checkpopint.

Resultado de la Proposición 1. *De acuerdo con las pruebas presentadas en Claims 1, 2 y 3, puede asegurarse que el protocolo satisface los requerimientos de seguridad necesarios (autenticidad, integridad y no repudio) para que las pruebas puedan considerarse válidas.*

Proposición 2. *El sistema de telepeaje presentado aquí preserva la privacidad de sus usuarios manteniendo su anonimato y evitando la trazabilidad de sus acciones.*

REIVINDICACIÓN 4. *El sistema garantiza el anonimato de sus usuarios honestos.*

PRUEBA. La información que el usuario debe transmitir para entrar al sistema es $\omega_{V_q} = (N_A, N_B, fing_{cert_{Chp}})$ y su firma. El Chp comprobará la firma con el certificado $cert_{V_q}$ que acompaña el mensaje del usuario. Este certificado (generado por el SE del *V* antes de la entrada a la *LEZ*) identifica el vehículo, pero esta información está protegida con un cifrado asimétrico usando la clave pública de la PA. Por tanto, el Chp puede comprobar la firma pero no identificar al vehículo. Posteriormente, el Chp genera y transmite al usuario la $\bar{\gamma_i}$. Con esta prueba el vehículo puede entrar a la *LEZ*. La $\bar{\gamma_i}$ no tiene más información del usuario que la contenida en ω_{V_q}. Esto significa que *V* entrará a la *LEZ* sin ser identificado.

Referente a la salida del sistema y, dejando de lado el sistema de pago que pueda ser utilizado (suponemos un sistema de pago que permita el anonimato), el usuario debe enviar al Chp de salida la siguiente información: $\omega_{V_q} = (ts, N_C, N_D, fing_{cert_{Chp}}$,*trans*). Al suponer que el pago es anónimo, no se puede identificar al usuario a través de *trans*. Tampoco, por la razón explicada en el anterior parágrafo se puede identificar al usuario a través de la firma de ω_{V_q}. Consecuentemente la salida y la entrada de usuarios honestos en una *LEZ* utilizando el sistema presentado aquí son anónimas.

REIVINDICACIÓN 5. *EL protocolo de telepeaje no permite rastrear o enlazar las operaciones de los vehículos.*

PRUEBA. La información que genera la ejecución del protocolo no permite enlazar las distintas entradas y salidas de las *LEZs* que pueda realizar un vehículo ya que el protocolo descrito en III.C se ejecuta cada vez que el *V* accede a una *LEZ*. Esto significa que el SE de vehículo genera un nuevo $cert_{V_q}$ cada nueva entrada. Este certificado es el único elemento que podría identificar al *V*. Ahora bien, teniendo en cuenta que el uso del certificado es único para cada entrada/salida, nadie puede relacionar el *V* de esta entrada/salida con otra ninguna otra entrada/salida.

La información que se podría repetir en otra entrada/salida del mismo *V* es el campo $cert_{V_q}.idS$ del certificado donde se encuentra la identidad del vehículo. Pero, tal y como se especifica en el protocolo, el $cert_{V_q}.idS$ esta calculado a partir de un cifrado probabilístico utilizando, por ejemplo, un sistema de padding OAEP que implica que el resultado de cada nueva operación de cifrado de las credenciales del *V* sea diferente.

Resultado de la Proposición 2. *El esquema de telepeaje presentado aquí preserva la privacidad de acuerdo con las argumentaciones 4 y 5: los usuarios pueden utilizar el sistema de forma anónima y cada uno de los usos no pueden ser relacionados entre si con respecto a la identidad de los vehículos.*

Proposición 3. *El sistema de telepeaje posee los requisitos antifraude en cuanto a la corrección y verificabilidad de las evidencias generadas en el protocolo.*

REIVINDICACIÓN 6.

Se puede identificar a los defraudadores gracias al sistema de revocación del anonimato que posee el protocolo.

PRUEBA. En caso que los usuarios no realicen de forma correcta la autenticación en la entrada y/o salida del sistema pueden perder el anonimato ya que el Chp realizará una foto al *V* y capturará la matricula. Esta información es enviada a la *PA* que actuará de la forma especificada en el protocolo de *Sanción*. En la ejecución del protocolo de *Sanción* la *PA* tiene la capacidad de identificar al usuario a través de la matricula del vehículo.

En caso que los usuarios no hayan realizado el pago de la forma correcta, la *SP* en el protocolo de *Verificación de pago* comprueba que la cantidad pagada se corresponda con la tarificación establecida en función de τ y de las emisiones de *V*. Si la verificación falla, se envía esta información a la *PA* para que el usuario sea sancionado. La *PA* ratifica la incidencia e identifica al usuario abriendo el campo del certificado $certV_q.idS$ con su clave secreta. La obtención de V_{id} permite identificar y sancionar la usuario deshonesto.

REIVINDICACIÓN 7.

La ejecución del protocolo genera evidencias para que un usuario honesto pueda guardar en su OBU *y pueda usarlas para comprobar o rebatir las acusaciones de fraude.*

PRUEBA. En el momento que un usuario es acusado de no realizar correctamente la autentificación se genera una pdi que registra la incidencia. El usuario puede ser acusado de usar un certificado $cert_{V_q}$ inapropiado o de realizar una firma $\omega_{\bar{V_q}}$ incorrecta. En ambos casos, durante el procedimiento de *Sanción* la *PA* se pone en contacto con él para que pueda aportar a la pruebas para rebatir la acusación.

Un usuario honesto podrá recuperar de su *OBU* un $cert_{V_q}$ que se corresponda con su vehículo (identificado por *mat*) o una $\omega_{\bar{V_q}}$ que la haya generado correctamente el *SE* con la ayuda de la *OBU* del usuario durante la el protocolo de entrada/salida de la *LEZ*. Cabe recordar que los requisitos del sistema establecen que la *OBU* de un vehículo tenga suficiente capacidad de almacenamiento para poder verificar las posibles acusaciones de fraude que se produzcan.

En caso de una incidencia de pago, el usuario debe demostrar que ha hecho el pago de acuerdo con los datos de $\bar{\rho}$ y $\bar{\gamma_o}$ (ambos firmados por el Chp). Por tanto, un usuario honesto

podrá recuperar estas informaciones de su *OBU* y remitirlas a la *PA* para rebatir la acusación.

Resultado de la Proposición 3. *El esquema de telepeaje presentado controla el fraude y puede identificar a los usuarios que lo han cometido realizando la correspondiente sanción. El protocolo permite también a los usuarios honestos obtener evidencias de su buen funcionamiento para desmentir posibles sanciones que se deban a algún tipo de funcionamiento incorrecto de los actores del sistema.*

V. Conclusiones y trabajo futuro

En este trabajo se ha presentado un sistema *ERP* para áreas urbanas que proporciona un sistema de control del fraude robusto con un elevado nivel de privacidad. Se controla la entrada y la salida de la LEZ de manera que se tarifica de forma justa y anónima. No obstante, si un usuario comete fraude es identificado mediante una foto, o gracias a la revocación de su privadidad.

Como trabajo futuro se considera la extensión del protocolo para considerar más de una LEZ, y su implementación para evaluar su aplicación práctica.

Agradecimientos

Este trabajo está parcialmente financiado por el Gobierno de España (a través de una beca FPI BES-2012-054780 y los proyectos CO-PRIVACY TIN2011-27076-C03-01, ARES-CONSOLIDER INGENIO 2010 CSD2007-00004 y BallotNext IPT-2012-0603-430000). Las opiniones, de los autores que pertenecen a la Cátedra UNESCO de privacidad de datos, expresadas en este artículo no reflejan necesariamente la posición de la UNESCO ni la comprometen.

Referencias

[1] "Resolución int/2836/2013." CVE-DOGC-B-14013017-2014. Núm 6541 - 15.1.2014.
[2] A. R.A.Popa, H.Balakrishnan, "Vpriv: protecting privacy in location-based vehicular services," in *SSYM'09*, 2009.
[3] S. J. X.Chen, G.Lenzini, "A group signature based electronic toll pricing system," in *ARES'12*, 2012.
[4] C. C. B. I. J.Balasch, A.Rial, "Pretp: Privacy-preserving electronic toll pricing," in *SSYM'10*, 2010.
[5] S. H. S.Meiklejohn, K.Mowery, "The phantom tollbooth: privacy-preserving electronic toll collection in the presence of driver collusion," in *SSYM'11*, 2011.
[6] E. I. J.Day, Y.Huang, "Spectre: spot-checked private ecash tolling at roadside," in *WPES '11*, 2011.
[7] B. F.Garcia, E.R.Verheul, "Cell-based privacy-friendly roadpricing," *Comput. Math. Appl.*, 2013.
[8] M.Bellare and P.Rogaway, "Optimal asymmetric encryption–how to encrypt with rsa," 1994.

Utilizando Certificados Implícitos para Asignar Identidades en Overlays P2P

Juan Caubet
Departamento de
Ingeniería Telemática (ENTEL)
Universitat Politècnica de Catalunya
Email: juan.caubet@entel.upc.edu

Jose L. Muñoz
Departamento de
Ingeniería Telemática (ENTEL)
Universitat Politècnica de Catalunya
Email: jose.munoz@entel.upc.edu

Oscar Esparza
Departamento de
Ingeniería Telemática (ENTEL)
Universitat Politècnica de Catalunya
Email: oscar.esparza@entel.upc.edu

***Resumen*—Desde hace años, la seguridad en las redes P2P estructuradas está siendo cuestionada, y por ello se han propuesto muchos trabajos con el objetivo de proporcionar enrutamiento seguro, sistemas de reputación, control de acceso, confidencialidad de los datos, etc. Sin embargo, el proceso de asignación de identidades se ha dejado casi totalmente olvidado. Estas redes están diseñadas para que cada usuario tenga un identificador único (nodeID), pero la mayoría de los sistemas existentes permiten que los usuarios puedan obtener un conjunto de ellos, e incluso seleccionarlos. Ambas actuaciones provocan problemas importantes de seguridad, ya que gracias a ello los usuarios pueden alterar el adecuado funcionamiento de la red. En este trabajo proponemos un protocolo de asignación de nodeIDs basado en la emisión de *certificados implícitos*. Nuestro propósito es proporcionar servicios de seguridad que permitan luchar contra la mayoría de las amenazas que sufren estas redes, con especial atención a la asignación de identidades. Este protocolo se basa en el uso de certificados y la generación conjunta de nodeIDs por parte la Autoridad de Certificación (CA) y el nuevo usuario.**

***Palabras clave*—Ataque Eclipse, Ataque Sybil, Overlay P2P, Gestión de Identidades, Certificados Implícitos**

I. Introducción

Las redes Peer-to-Peer (P2P) estructuradas, de aquí en adelante overlays P2P, aparecieron hace unos años para resolver problemas de enrutamiento en grandes infraestructuras distribuidas, incluso a nivel de Internet; ya que pueden proporcionar escalabilidad, tolerancia a fallos, auto-organización y baja latencia. Por ello, y según el estudio anual "Cisco Visual Networking Index (VNI) Forecast" [1], el tráfico P2P representaba alrededor del 30 % del tráfico IP global en 2011 y crecerá con una tasa de crecimiento anual compuesto (CAGR) del 23 % de 2010 a 2015. Sin embargo, las overlays P2P no están siendo ampliamente utilizadas por aplicaciones comerciales, ya que presentan importantes problemas de seguridad. Todos sabemos que hoy en día la mayor parte de estas redes funcionan razonablemente bien sin necesidad de profundizar en la seguridad, pero no hay que perder de vista que proporcionan servicios gratuitos; y por ello los usuarios están dispuestos a asumir ciertas deficiencias (no garantizan una calidad de servicio (QoS) mínima), e incluso algún que otro riesgo.

Las aplicaciones P2P de video streaming son un buen ejemplo de aplicaciones comerciales que necesitan especial atención en la seguridad. La transmisión de video en redes P2P surgió como una evolución al intercambio de contenidos multimedia mediante descarga. Su poder para dar cabida a millones de usuarios, junto con su capacidad de resistencia al dinamismo, su fiabilidad y su bajo coste, son algunas de las razones por las que las redes P2P están siendo cada vez más utilizadas por este tipo de aplicaciones. El ejemplo es que las aplicaciones de video bajo demanda (VoD) producirán tres veces más tráfico en 2015 del que produjeron en 2011 [1]. SopCast, PPTV, CoolStreaming, TVUnetworks y Zattoo son algunas de las muchas aplicaciones de streaming de video que se han desarrollado hasta el momento. Sin embargo, la mayoría de ellas son plataformas propietarias, las cuales utilizan la segunda generación de redes P2P o distribuyen contenidos con un control de acceso deficiente, y poca o nula seguridad.

Debido a sus características, las overlays P2P presentan vulnerabilidad ante ciertos ataques, lo cual debe ser solucionado si queremos utilizar estas redes para implementar aplicaciones comerciales (como servicios de VoD bajo suscripción). Además, esta medida ayudaría a que los usuarios tengan más confianza en las redes P2P, ya que a menudo se piensa que son inseguras por naturaleza.

Las overlays P2P han sido analizadas en profundidad para garantizar su escalabilidad y eficiencia. Sin embargo, pocos mecanismos de seguridad se están utilizando en la actualidad. La mayoría de estas redes asumen que los nodos tienen un comportamiento honesto, pero este supuesto no es aceptable en entornos abiertos. La existencia de nodos anónimos y la falta de una autoridad centralizada capaz de controlar (y castigar) a los nodos, hace que estos sistemas sean vulnerables frente a comportamientos egoístas y maliciosos. Y desafortunadamente estos comportamientos no pueden ser evitados únicamente mediante el uso de los servicios básicos de seguridad. Las overlays P2P también deben seguir las primitivas de enrutamiento seguro descritas por Wallach en [2], que son: (1) el mantenimiento seguro de las tablas de enrutamiento, (2) el enrutamiento seguro de mensajes, y (3) la asignación segura de los identificadores de nodo (nodeIDs). Sin embargo, los dos primeros paradigmas dependen directamente del tercero. Si los nodeIDs pueden ser elegidos por los usuarios sin ningún tipo de control, podemos tener problemas de seguridad y funcionamiento. Desafortunadamente, hasta ahora se ha

prestado poca atención a la forma en que los nodeIDs deben ser construidos, o cómo hacer los mecanismos de control de acceso más robusto. Al igual que cualquier otra red, las overlays P2P requieren de un control de acceso eficiente para prevenir el acceso de posibles atacantes a la red. Pero además, éstas deberían disponer de un sistema de asignación de identidades robusto, con el fin de mejorar la confianza de los usuarios en estas redes y que así puedan ser mayormente utilizadas por aplicaciones comerciales.

Por ello proponemos el uso de certificados digitales y un nuevo protocolo de asignación de identidades, el cual aprovecha la emisión de estos certificados. Más concretamente, proponemos el uso de *certificados implícitos* [3], [4], los cuales presentan ciertas ventajas sobre los certificados tradicionales (*certificados explícitos*). Los certificados implícitos tienen un menor tamaño, ya que no incluyen la firma de la entidad emisora, y pueden ser verificados más rápido, ya que requiere menos tiempo de cálculo la reconstrucción de una clave pública que verificar una firma digital. Por otra parte, la generación de estos certificados nos permite construir nodeIDs de forma segura. Para ello utilizamos las claves públicas, las cuales son construidas conjuntamente por los usuarios y la autoridad de certificación (CA), y así minimizamos el impacto de los ataques Eclipse [5], entre otros.

El resto del artículo está organizado de la siguiente forma: La sección II explica algunos de los problemas existentes en las overlays P2P relacionados con las identidades. La sección III presenta algunas propuestas que intentan prevenir, detectar y/o limitar los problemas de identidad experimentados en estas redes. La sección IV explica que son los esquemas de compromiso y describe un esquema de certificados implícitos basado en curvas elípticas. La sección V presenta nuestro protocolo de asignación de identidades para una overlay P2P. Y por último, en la sección VI se extraen algunas conclusiones.

II. Problemas de Identidad en Overlays P2P

La mayoría de las overlays P2P están implementadas utilizando tablas de hash distribuidas (DHTs), que almacenan pares $\{clave, valor\}$ junto con los nodeIDs creando un espacio virtual. Un *valor* puede ser un recurso (por ejemplo, un archivo), o la forma de llegar a él (un puntero), y la *clave* asociada indica su ubicación. La DHT se divide en subtablas, las cuales corresponden a una zona determinada del espacio virtual y van siendo asignadas a los diferentes nodos. Así cada nodo es responsable de una zona, y por lo tanto es responsable de los pares {clave,valor} contenidos en esa zona (almacenando mensajes de contenido y enrutamiento). Por lo general, una zona es asignada a un nodo cuyo nodeID es numéricamente cercano a los valores de las claves almacenadas en la subtabla correspondiente. Por lo tanto, la ubicación de los nodos en el espacio virtual está directamente relacionada con sus nodeIDs. Y desafortunadamente, en la mayoría de las overlays P2P actuales estos identificadores son generados por los usuarios localmente, lo que significa que pueden elegir sus nodeIDs y consecuentemente su ubicación en la overlay.

Los usuarios en la red CAN [6] son identificados por la zona que tienen asignada dentro del espacio virtual, zonas seleccionadas por ellos mismos. En las redes Chord [7] y Kademlia [8], los nodeIDs son generados por los usuarios utilizando una función hash sobre sus direcciones IP. En Pastry [9] los nodeIDs son asignados al azar por el software del cliente. Y de manera similar en otras overlays P2P.

Varios problemas relacionados con las identidades surgen de la asignación descontrolada de los nodeIDs: Ataques Sybil, ataques Eclipse, ataques Man-in-the-Middle (MITM), la presencia de whitewashers, etc. A continuación se describen algunas de las amenazas más importantes.

II-A. El ataque Sybil

La gestión de múltiples nodeIDs (Sybils) por parte del mismo nodo se conoce como ataque Sybil [10]. Llevando a cabo este ataque, un usuario malintencionado puede aumentar su presencia dentro de la overlay simulando artificialmente la existencia de varios nodos. Por lo tanto, el atacante que puede manejar un grupo de nodos puede alterar el funcionamiento de la red, o simplemente mejorar su reputación.

II-B. El ataque Eclipse

El ataque Eclipse [5] pretende alterar la información de enrutamiento de un nodo (o grupo de nodos) objetivo para aislarlo del resto de la overlay. El atacante interceptará los mensajes dirigidos a dicho nodo (o grupo) mediante un conjunto de nodos confabulados (o Sybils) que se encuentran cercanos al objetivo con el fin de controlar sus comunicaciones.

II-C. El Ataque Man-In-The-Middle (MITM)

Como su nombre indica, en este ataque el atacante se sitúa entre dos nodos con el propósito de espiar sus comunicaciones, o incluso manipularlas. Por lo general, en las redes P2P, el objetivo de estos atacantes es robar nodeIDs y/o generar información falsa. Por lo tanto, si tenemos en cuenta el tipo de enrutamiento de estas redes y permitimos que los nodeIDs sean seleccionados por los usuarios sin ningún control, no hay duda de que estas redes son extremadamente vulnerables a ataques MITM.

II-D. Otros Problemas

La eficiencia de los algoritmos de enrutamiento se basa en la uniforme distribución de los nodeIDs. Por lo tanto, el rendimiento de una overlay puede ser globalmente degradada si la mayoría de los nodeIDs pertenecen a una sola zona del espacio virtual. Y desafortunadamente, si los nodeIDs pueden ser seleccionados por los usuarios, nadie tendrá la seguridad de que los identificadores van a estar distribuidos uniformemente.

Otra amenaza a la seguridad relacionada con los nodeIDs es la presencia de whitewashers (nodos que intencionadamente abandonan la red y vuelven a entrar en ella con un nuevo nodeID con la intención de limpiar su mala reputación [11]). Los sistemas de reputación pueden ser utilizados para prevenir comportamientos maliciosos y promover la colaboración entre los nodos. Sin embargo, la eficacia de estos sistemas depende de la estabilidad de los nodeIDs.

III. Estado del Arte

Douceur [10] fue el primero en tratar el ataque Sybil en overlays P2P y comentar la imposibilidad de saber si dos nodos son gestionados por dos usuarios diferentes, o si en realidad lo hace uno solo; incluso recabando información de otros nodos de la red. De esta forma concluye que una entidad de confianza que certifique los nodeIDs es la única solución para evitar por completo el ataque Sybil en estas redes. Sin embargo, también sugiere el uso de métodos que añadan un coste computacional al proceso de obtención de nodeIDs para mitigar el ataque. Siguiendo esta línea, hasta la fecha se han propuesto muchas alternativas.

En [12], Castro et al. proponen dos formas centralizadas de generar nodeIDs. La primera de ellas es delegar el problema a un conjunto de entidades de confianza, las cuales firman los certificados vinculándolos con un nodeID aleatorio, una clave pública y la dirección IP del usuario. La segunda propuesta consiste en cobrar dinero por los certificados, u obligar a los usuarios a vincular su identidad real con los nodeIDs. Srivatsa y Liu proponen el uso de certificados con un tiempo de vida limitado y emitidos por una CA, el cual también vincula los certificados a nodeIDs aleatorios [13]. En [14], Butler et al. consideran el uso del encriptado basado en identidad (IBE), donde las claves públicas son derivadas directamente de los nodeIDs. Los nodeIDs son generados aleatoriamente por una CA y la autenticación de los nodos se lleva a cabo a través de un proceso de *callback* utilizando la dirección IP del usuario. En [15], Aiello et al. proponen, por una parte introducir la interacción humana en la fase de autenticación utilizando el protocolo OpenID, y por otra parte utilizar una entidad de confianza que vincule la identidad real del usuario con su clave pública y con un nodeID aleatorio para generar un LikirID. En [16], Rowaihy et al. han propuesto un mecanismo de puzzles criptográficos para limitar el ataque Sybil. Proponen un sistema de control de admisión utilizando una estructura jerárquica autoorganizada de nodos y una cadena de puzzles criptográficos. Ellos explotan dicha estructura jerárquica para distribuir la carga y aumentar la capacidad de resistencia a los ataques dirigidos, y actualizan los puzzles con frecuencia para así evitar la precomputación. En [12], [17], [18], sus autores también utilizan puzzles criptográficos para limitar el ataque Sybil.

IV. Background

IV-A. *Esquemas de compromiso (Commitment Schemes)*

Un esquema de compromiso es un protocolo interactivo entre dos participantes (Emisor y Receptor), destinado a ocultar temporalmente un valor que ya no debe ser cambiado. Es decir, el Emisor se compromete a utilizar un valor, el cual ha de permanecer temporalmente oculto para el Receptor. Estos sistemas suelen consistir de dos fases:

1. Fase 1 (Compromiso): el Emisor se compromete a utilizar un determinado valor.
2. Fase 2 (Revelación): el Emisor prueba al Receptor que el valor no ha sido cambiado desde entonces.

Estos esquemas son primitivas muy útiles en criptografía y siempre deben cumplir con dos propiedades: *Vinculación* y *Ocultación*. La Vinculación asegura que en la fase de Revelación un compromiso sólo pueda revelar con éxito un valor (unicidad). La Ocultación garantiza que la fase de Compromiso no revela ninguna información sobre el valor oculto (secreto prefecto). Tanto la Vinculación como la Ocultación pueden ser garantizadas (estadística o computacionalmente) en función de la potencia de cálculo necesaria para romperlas. Estos esquemas son aplicados en protocolos tales como las pruebas de conocimiento cero (Zero-knowledge), la computación multiparte, las subastas digitales o el comercio electrónico.

En este artículo nosotros definimos un nuevo esquema de compromiso basado en la criptografía basada en curvas elípticas (ECC) con el fin de mejorar la seguridad en un protocolo de emisión de certificados implícitos.

IV-B. *Certificados Implícitos*

Un certificado estándar contiene explícitamente la clave pública del usuario y la firma de la CA que ha emitido dicho certificado, junto con otra información adicional (número de serie, período de validez, identidad del emisor, identidad del usuario, etc.). Un certificado implícito [3], [4] no contiene la clave pública del usuario ni la firma de la CA. En lugar de ello contiene la información necesaria para calcular su clave pública asociada, un parámetro de reconstrucción.

Por lo tanto, un certificado implícito es simplemente un par (I, Z), donde I denota la información incluida en el certificado y Z denota el parámetro de reconstrucción. Los certificados implícitos tienen una longitud más corta que los explícitos y proporcionan así una alternativa más eficiente.

Antes de validar la firma de un emisor, cualquier receptor de un certificado implícito debe reconstruir la clave pública asociada utilizando Z y la clave pública de la CA emisora. De la misma manera que con los certificados explícitos, el receptor debe confiar en la CA y disponer de su clave pública para tener así la seguridad de que la clave reconstruida ha sido emitida por dicha CA. Con los certificados explícitos, el receptor verifica la firma del certificado con la clave pública de la CA, y a partir de ese momento puede estar seguro de que la clave contenida en el certificado pertenece a un determinado usuario y ha sido emitida por esa CA. Sin embargo, validar únicamente el certificado no es suficiente para autenticar a un usuario. Por lo tanto, para autenticar a un usuario, éste debe demostrar el conocimiento de la clave privada asociada utilizando un protocolo criptográfico seguro. Y lo mismo aplica a los certificados implícitos, donde la autenticación de una clave pública y la autenticación de pertenencia a un usuario no son separables.

La figura 1 ilustra el esquema de emisión de certificados implícitos "Elliptic Curve Qu-Vanstone" (ECQV) [4], propuesto por el Standards for Efficient Cryptography Group (SECG). En él un usuario X solicita un nuevo certificado enviando un punto aleatorio dentro de una curva elíptica (N_X), el cual es utilizado por la CA para generar el parámetro de reconstrucción de su nueva clave pública ($Z = N_X + N$). Una

vez calculado Z, la CA calcula el valor del hash del certificado ($h = H(I||Z)$) y la firma (s). Finalmente X recibe su nuevo certificado (Z, I) y su firma (s), genera su nuevo par de claves criptográficas (la clave privada d_X y la clave pública Q_X) utilizando la clave pública de la CA (Q_{CA}).

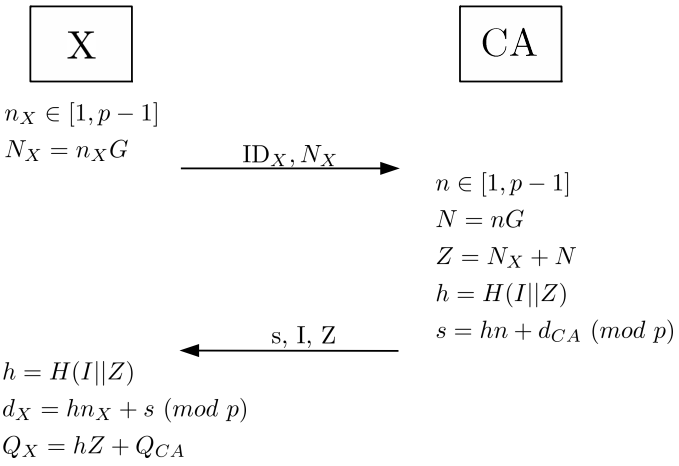


Figura 1. Protocolo de Emisión de Certificados Implícitos ECQV.

En este artículo proponemos una serie de modificaciones sobre este sistema de emisión de certificados con el fin de proponer un nuevo sistema de gestión de identidades seguro.

V. Protocolo de Asignación de Identidades

Nuestro protocolo asigna nodeIDs de forma segura y eficiente aprovechando la emisión de certificados implícitos. Estos certificados proporcionan identificación digital para autenticar usuarios, soporte a la criptografía de clave pública, etc.; pero también presentan ciertas ventajas sobre los certificados tradicionales. En este protocolo, los NodeIDs se calculan utilizando una función de hash sobre la clave pública de los usuarios, pero a diferencia de otras propuestas, estas claves públicas son generadas bajo la supervisión y participación de una CA, la cual no conoce las claves privadas asociadas. El esquema de emisión de certificados ECQV [4] ha sido modificado con el fin de garantizar que ninguna de las dos partes involucradas en el proceso tenga la capacidad de elegir el valor de la clave pública emitida.

V-A. Suposiciones y Clarificaciones

Con el fin de adaptar el esquema ECQV a nuestras necesidades, hemos definido un nuevo esquema de compromiso basado en curvas elípticas. Este esquema ha sido construido inspirándonos en el cifrado Exclusive-OR (XOR) y suponiendo que un emisor S posee una clave privada d_S y una clave pública $Q_S = d_S G$, donde G es el generador de la curva elíptica. La figura 2 describe el protocolo en detalle, donde u es un número aleatorio, U es el punto de la curva elíptica asociado a u, c es el valor de compromiso y v es el valor elegido por S. En la primera fase, S se compromete a utilizar un valor v enviando los valores c y U a R. Y en la segunda fase, S revela el valor del número aleatorio utilizado (u) y R chequea que u realmente fue utilizado para generar U. Finalmente R calcula el valor de v utilizando c.

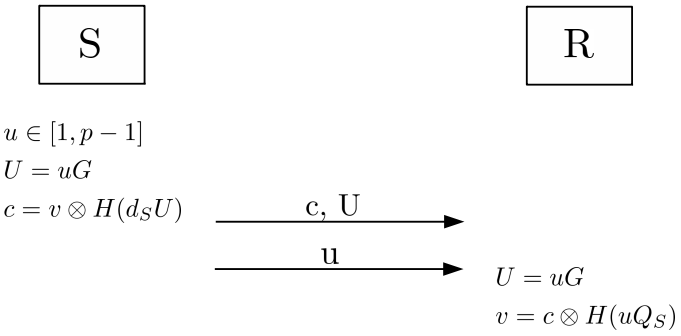


Figura 2. Nuevo Esquema de Compromiso.

En este esquema de compromiso se cumplen las dos principales propiedades de seguridad que requieren este tipo de protocolos: *Vinculación* y *Ocultación*. La Vinculación es segura, ya que dos valores diferentes de u no pueden dar como resultado el mismo valor correcto de v. Y la Ocultación es computacional, ya que dados los valores c, U y G, un atacante puede resolver el problema del logaritmo discreto en una curva elíptica (ECDLP) para obtener el valor aleatorio u, para luego para calcular v.

La modificación realizada en el esquema ECQV ha consistido en añadir el esquema de compromiso anteriormente explicado. Con ello evitamos que la CA pueda elegir el valor del parámetro de reconstrucción (Z) una vez conocido el valor enviado por el usuario (N_X) durante el proceso de generación de certificados. Esto implica que ahora la CA debe seleccionar N y enviar un compromiso (c, U) al usuario antes de recibir N_X. Así, una vez que el usuario ha recibido su nuevo certificado, él será capaz de verificar que el certificado se ha generado utilizando el valor al que se comprometió la CA.

V-B. Especificaciones

En esta sección describimos el protocolo en detalle; la información intercambiada, la forma de intercambiarla, los mecanismos de seguridad utilizados, etc. La figura 3 muestra la información intercambiada por ambas partes y las operaciones llevadas a cabo, pero sin tener en cuenta las operaciones de cifrado y firma. La tabla I presenta un resumen global de la notación utilizada a lo largo de esta sección.

V-B1. Paso 1: Cada vez que un usuario quiere unirse a la overlay P2P contacta con la CA enviando un "HELLO MESSAGE", el cual contiene las identidades (ID_X y ID_{CA}), una marca temporal (t_X), también utilizada como identificador de petición, y el certificado del usuario (C_X). Este mensaje es firmado por el usuario utilizando su clave privada en el mundo real, y cifrado utilizando la clave pública de la CA.

$$HELLO\ MESSAGE,\ X \rightarrow CA:$$
$$\{C_X, \{ID_X, ID_{CA}, t_X\}_{d_X}\}_{Q_{CA}}$$

Tabla I
NOTACIÓN

p	El orden del cuerpo finito $\mathbb{F}_p$.
G	El generador de la curva elíptica definida sobre $\mathbb{F}_p$ ($E(\mathbb{F}_p)$).
ID_X	La identidad del nuevo usuario X.
ID_{CA}	La identidad de la CA.
P_X	El seudónimo de X dentro de la overlay (nodeID).
C_X	El certificado digital de X en el mundo real.
d_X	La clave privada de X en el mundo real.
Q_X	La clave pública de X en el mundo real.
d_{Xo}	La clave privada de X dentro de la overlay.
Q_{Xo}	La clave pública de X dentro de la overlay.
d_{CA}	La clave privada de la CA.
Q_{CA}	La clave pública de la CA.
t_X	La marca temporal generada por el nuevo usuario (identificador de petición).
I	La información incluida en el certificado de X.
Z	El parámetro de reconstrucción de la clave pública de X.
h	El resumen del nuevo certificado ($I\|\|Z$).
s	La firma de la CA para el nuevo certificado de X.
$n,\ u,\ n_X$	Parámetros privados de la petición de X ($\in [1,\ p-1]$).
$N,\ U,\ N_X$	Parámetros públicos de la petición de X ($\in E(\mathbb{F}_p)$).
c	El valor de compromiso.
$H(m)$	Una función de hash sobre un mensaje m.
$i \rightarrow j:$	El envío de un mensaje de la entidad i a la entidad j.
$\{m\}_Q$	El texto cifrado de un mensaje m utilizando la clave pública Q.
$\{m\}_d$	La firma sobre un mensaje m utilizando la clave privada d.

V-B2. Paso 2: Cada vez que un nuevo usuario contacta con la CA, ésta genera dos parámetros privados ($\{n,\ u\} \in [1,\ p-1]$), sus respectivos puntos en la curva elíptica ($N = nG$ y $U = uG$) y calcula $c = N \otimes H(d_{CA}U)$. Finalmente envía c y U a X, todo firmado y cifrado junto con las identidades y la marca temporal.

$$ACCEPT\ MESSAGE,\ CA \rightarrow X:$$
$$\{\{ID_{CA}, ID_X, t_X, c, U\}_{d_{CA}}\}_{Q_X}$$

V-B3. Paso 3: X recibe el "ACCEPT MESSAGE" y genera un parámetro privado $n_X \in [1,\ p-1]$ y su punto asociado $N_X = n_X G$. Después envía N_X a la CA, firmado y cifrado junto con las identidades y la marca temporal.

$$REQUEST\ MESSAGE,\ X \rightarrow CA:$$
$$\{\{ID_X, ID_{CA}, t_X, N_X\}_{d_X}\}_{Q_{CA}}$$

V-B4. Paso 4: La CA recibe el "REQUEST MESSAGE", calcula el parámetro de reconstrucción ($Z = N_X + N$) y el valor de hash de ese parámetro concatenado con I ($h = H(I||Z)$). Después firma el certificado ($s = hn + d_{CA}\ mod\ p$) y le proporciona a X los valores $\{s, r, I, Z\}$, todos firmados y cifrados junto con las identidades y la marca temporal.

$$RESPONSE\ MESSAGE,\ CA \rightarrow X:$$
$$\{\{ID_{CA}, ID_X, t_X, s, u, I, Z\}_{d_{CA}}\}_{Q_X}$$

Note que una vez que Z y h han sido calculados, si $Q_{Xo} = hZ + Q_{CA} = \mathcal{O}$, la CA le pide al usuario que le mande un nuevo parámetro y repite el proceso.

V-B5. Paso 5: X recibe su nuevo certificado y la firma de la CA ("RESPONSE MESSAGE"), y calcula $U' = uG$ (y compara éste con U), $N = Z - N_X$ y $N' = H(uQ_{CA}) \otimes c$ para verificar que la CA ha utilizado el valor inicial n; si no es así cancela el proceso. Después X genera su clave privada $d_{Xo} = hn_X + s\ mod\ p$ y su clave pública $Q_{Xo} = d_{Xo}G$, y calcula su nuevo nodeID como el valor de hash de Q_{Xo} ($P_X = H(Q_{Xo})$). Finalmente envía su nodeID junto con las identidades y la marca temporal, todo firmado y cifrado.

$$CONFIRMATION\ MESSAGE,\ X \rightarrow CA:$$
$$\{\{ID_X, ID_{CA}, t_X, P_X\}_{d_X}\}_{Q_{CA}}$$

V-B6. Generación de la Clave Pública: Cada vez que un usuario recibe un mensaje, éste necesita generar la clave pública del emisor para poder autenticarlo y verificar la firma del mensaje. Para ello utiliza su certificado implícito, el cual incluye la información del certificado (I) y el parámetro de reconstrucción (Z). Finalmente sigue los siguientes pasos:

1. Calcula el parámetro $h = H(I||Z)$.
2. Genera la clave pública del emisor $Q_{Xo} = hZ + Q_{CA}$.
3. Verifica la firma del mensaje utilizando Q_{Xo}.

Nótese que dicha verificación se cumplirá porque:

$Q_{Xo} = d_{Xo}G = hn_XG + sG = hn_XG + hnG + d_{CA}G = hN_X + hN + Q_{CA} = h(N_X + N) + Q_{CA} = hZ + Q_{CA}$.

V-B7. Validación del nodeID: Cada vez que un nodo recibe información de otro nodo (contenidos o información de enrutamiento) debe validar su nodeID. Para ello, el nodo sólo tiene que calcular el hash de su clave pública ($P_X = H(Q_{Xo})$) y compararlo con el nodeID utilizado.

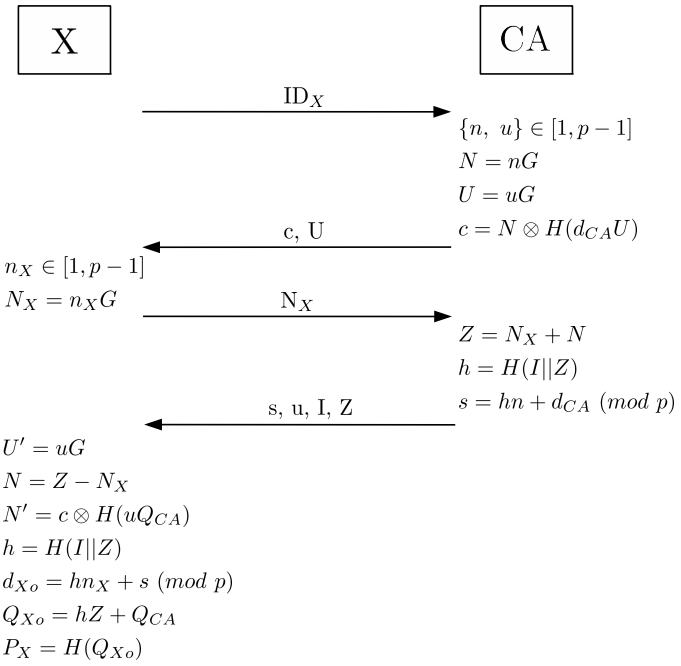


Figura 3. Esquema de Generación de nodeIDs/Certificados.

VI. Conclusiones

La vulnerabilidad a ciertos ataques es un fuerte obstáculo para el desarrollo de aplicaciones comerciales en las overlays P2P. En este artículo se ha propuesto un esquema seguro de asignación de identidades con el objetivo de resolver algunas de estas vulnerabilidades y convertir estas redes en una potente plataforma para aplicaciones comerciales. Nuestro protocolo genera nodeIDs de forma segura y anónima, sin afectar al funcionamiento actual de la red. Éste no permite ni que los usuarios seleccionen sus nodeIDs ni que la CA pueda seleccionarlos por ellos, y garantiza que los nodos sean ubicados en el espacio virtual de forma pseudo-aleatoria (uniformemente). Finalmente, hay que tener en cuenta que cualquier sistema de seguridad implica un compromiso entre el nivel de seguridad y el rendimiento de la red. Pero en nuestro caso, y teniendo en cuenta que un usuario sólo ejecutaría el protocolo la primera vez que quiere unirse a la red, la calidad experimentada por el usuario (QoE) no se verá afectada. En cuanto a la seguridad, nuestra propuesta sólo tiene una debilidad; debemos confiar en la CA. Pero hay que tener en cuenta que utilizar una CA es la única forma de evitar 100 % ciertos ataques (ataque Sybil, ataque Eclipse, etc.). El trabajo futuro se centrará en proponer un sistema de gestión de identidades que proporcione trazabilidad de usuarios y revocación de certificados y nodeIDs.

Reconocimientos

Este trabajo ha sido parcialmente subvencionado por la Secretaría de Estado de Investigación, Desarrollo e Innovación bajo los proyectos SERVET TEC2011-26452 y CONSOLIDER CSD2007-00004 (ARES), y por la Generalitat de Catalunya bajo la ayuda 2009 SGR-1362 para grupos consolidados.

Referencias

[1] Cisco Systems, Inc, "Cisco Visual Networking Index: Forecast and Methodology, 2011-2016," http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html.

[2] D. S. Wallach, "A Survey of Peer-to-Peer Security Issues," in *Proceedings of the Mext-NSF-JSPS international conference on Software security: theories and systems*, ser. ISSS'02. Tokyo, Japan: Springer-Verlag Berlin, Heidelberg, 2002, pp. 42–57.

[3] D. R. Brown, R. Gallant, and S. A. Vanstone, "Provably Secure Implicit Certificate Schemes," in *Financial Cryptography*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, vol. 2339, pp. 156–165.

[4] C. Research, "Standards for Efficient Cryptography 4 (SEC 4): Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV)," November 2013, version 1.1.

[5] R. Fantacci, L. Maccari, M. Rosi, L. Chisci, L. M. Aiello, and M. Milanesio, "Avoiding Eclipse Attacks on Kad/Kademlia: An Identity Based Approach," in *Proceedings of the IEEE International Conference on Communications*, ser. ICC'09. IEEE Press, june 2009, pp. 983–987.

[6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM)*, San Diego, CA, USA, 2001, pp. 161–172.

[7] I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishman, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM)*, San Diego, CA, USA, 2001, pp. 149–160.

[8] P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *Proceedings of the 1st International Workshop on Peer-to Peer Systems*, ser. IPTPS'02, march 2002, pp. 53–65.

[9] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," in *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms*, 2001, pp. 329–350.

[10] J. R. Douceur, "The Sybil Attack," in *Proceedings of the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS'02. London, UK: Springer-Verlag, 2002, pp. 251–260.

[11] S. Marti and H. Garcia-Molina, "Taxonomy of Trust: Categorizing P2P Reputation Systems," *Computer Networks*, vol. 50, no. 4, pp. 472–84, 2006.

[12] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure Routing for Structured Peer-to-peer Overlay Networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 299–314, december 2002.

[13] M. Srivatsa and L. Liu, "Vulnerabilities and Security Threats in Structured Overlay Networks: A Quantitative Analysis," in *Proceedings of the 20th Annual Computer Security Applications Conference*, december 2004, pp. 252–261.

[14] K. R. Butler, S. Ryu, P. Traynor, and P. D. McDaniel, "Leveraging Identity-Based Cryptography for Node ID Assignment in Structured P2P Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 12, pp. 1803–1815, december 2009.

[15] L. M. Aiello, M. Milanesio, G. Ruffo, and R. Schifanella, "An identity-based approach to secure P2P applications with Likir," *Peer-to-Peer Networking and Applications*, vol. 4, pp. 420–438, 2011.

[16] H. Rowaihy, W. Enck, P. McDaniel, and T. L. Porta, "Limiting Sybil Attacks in Structured P2P Networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications*, Anchorage, Alaska, USA, may 2007, pp. 2596–2600.

[17] W. L. D. C. Cordeiro, F. R. Santos, G. H. Mauch, M. P. Barcelos, and L. P. Gaspary, "Identity management based on adaptive puzzles to protect P2P systems from Sybil attacks," *Comput. Netw.*, vol. 56, no. 11, pp. 2569–2589, july 2012.

[18] C. Lu, "Detection and Defense of Identity Attacks in P2P Network," in *Advances in Computation and Intelligence*, ser. Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg, 2009, vol. 5821, pp. 500–507.

Cálculo Privado de Distancias entre Funciones de Preferencia

Alberto Blanco-Justicia
Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili
Email: alberto.blanco@urv.cat

Josep Domingo-Ferrer
Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili
Email: josep.domingo@urv.cat

Oriol Farràs
Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili
Email: oriol.farras@urv.cat

David Sánchez
Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili
Email: david.sanchez@urv.cat

***Resumen*—Consideremos el siguiente escenario: dos entidades quieren saber el grado de semejanza que hay entre ellas. Sus perfiles se pueden describir a través de funciones de preferencia, y querrían calcular la distancia entre estas funciones sin tener que revelarlas. Este escenario parece de especial relevancia en el contexto de las redes sociales, políticas o empresariales, cuando uno desea encontrar amigos o socios con intereses parecidos sin tener que revelar sus intereses a nadie. En este trabajo, proporcionamos protocolos que resuelven el problema anterior para distintos tipos de funciones. Los experimentos, además, demuestran que es posible realizar estos cálculos de manera privada, eficiente y sin causar reducciones significativas en la precisión de las distancias calculadas manteniendo, por tanto, su utilidad.**

***Palabras clave*—Cálculo privado de distancias, privacidad, redes sociales, funciones de utilidad, preferencias, perfiles de usuario, emparejamiento privado.**

I. Introducción

La timidez puede tener un componente racional. Llegar a conocer a un extraño requiere habitualmente que le revelemos parte de nuestra información privada. De hecho, en una relación justa hay normalmente un intercambio mutuo de información, en la que cada una de las partes debe revelar algo a la otra con tal de aprender algo. Una manera de preservar la privacidad y limitar riesgos sería que ambas partes pudiesen determinar si tienen intereses parecidos sin ninguna revelación *a priori*. Evidentemente, cuanto más semejantes resulten ser sus intereses, mayores serán las revelaciones mutuas *a posteriori*: en el caso extremo en que sus intereses estuvieran a distancia 0, se produciría una revelación total de sus intereses.

En términos de teoría de juegos, el anterior problema se puede expresar como dos jugadores interesados en determinar cuán cerca están sus funciones de utilidad sin revelar esas funciones de utilidad al otro jugador. En consecuencia, esto permitiría formar coaliciones con intereses homogéneos sin ninguna revelación *a priori*.

Encontrar una solución a este problema podría resultar muy relevante para resolver varias situaciones reales:

- En redes sociales, los usuarios podrían encontrar amigos o usuarios a los que seguir que compartan sus intereses, sin ser forzados a revelar sus propios intereses privados (p.e. religión, orientación sexual, condición de salud, etc.). Por ejemplo, actualmente en la red social `PatientsLikeMe` [12] los usuarios tienen que revelar sus enfermedades para encontrar a otros usuarios con condiciones médicas parecidas. La pérdida de privacidad es evidente y podría ser mitigada por nuestra propuesta.
- Los ataques de *grooming* podrían ser mitigados significativamente con nuestra propuesta. Nótese que el agresor tendría que adivinar los intereses de su víctima para poder llegar a estar entre sus amistades.
- Modelar consumidores con perfiles específicos también sería posible. Las empresas podrían crear un usuario falso en redes sociales con el tipo de perfil de los consumidores que busca. De esta manera, las empresas podrían identificar comunidades de potenciales clientes con el perfil deseado, sin entrometerse en la privacidad de los usuarios que no encajan en el perfil que buscan.
- En tratos comerciales, las partes podrían determinar si la importancia que asignan a cierta colección de bienes es similar a la asignada por otras, sin revelar sus estrategias comerciales. Por ejemplo, en algunos casos una compañía podría estar interesada en asociarse con empresas con intereses distintos, para formar alianzas complementarias, en lugar de asociarse a empresas con intereses demasiado parecidos, que podrían ser vistas como competidoras.
- En procesos de contratación, empresas y candidatos serían capaces de determinar confidencialmente hasta qué punto la visión corporativa de la empresa es compartida por cada candidato. Gracias a un mecanismo que preserve la privacidad, se podrían incluir en la evaluación un gran número de factores distintos, sin que la empresa revele sus objetivos estratégicos a los candidatos no elegidos ni éstos revelen sus opiniones.

I-A. *Contribución y estructura del artículo*

En este artículo presentamos protocolos que permiten calcular privadamente la distancia entre varios tipos de funciones. A continuación, mostramos los resultados experimentales que demuestran que preservar la privacidad no causa una distorsión significativa en las distancias calculadas.

La sección II define diversos casos de cálculos privados de

distancias entre funciones, dependiendo de la naturaleza de la función y del tipo de distancia que se considera. La sección III describe un protocolo para calcular las distancias entre funciones, preservando la privacidad, basada en la intersección de conjuntos. La sección IV muestra los resultados del trabajo experimental. La sección V describe trabajos relacionados. Las conclusiones y trabajo futuro se resumen en la sección VI.

II. Taxonomía de cálculos de la distancia entre funciones

El protocolo para calcular de manera privada la distancia entre dos funciones depende principalmente de la naturaleza de dichas funciones y de la manera en que se miden las distancias entre ellas. A continuación discutimos varios de estos casos.

II-A. Caso A: número de preferencias cualitativas en común

En este primer caso, los intereses o preferencias de cada una de las partes se representan como conjuntos de valores booleanos relacionados con varios temas independientes. Por ejemplo, en las redes sociales como Facebook, se les pide a los usuarios que den sus opiniones sobre diversos temas en forma de "me gusta". En `PatientsLikeMe` [12], los usuarios detallan sus historiales médicos como selecciones binarias entre conjuntos de alternativas (enfermedades, síntomas, etc.).

De este modo, consideramos las preferencias del usuario o entidad (su perfil) como un conjunto que contiene sus opiniones y/o detalles personales. Definimos este conjunto como X para el primer jugador $\mathcal{C}$ e Y para el segundo, $\mathcal{S}$. La distancia entre los intereses de $\mathcal{C}$ y $\mathcal{S}$ se puede calcular como el inverso multiplicativo del tamaño de la intersección de X e Y, es decir $1/|X \cap Y|$, siempre que la intersección sea no nula. Si lo fuese, podríamos considerar que la distancia es ∞.

Evidentemente, cuanto más coincidencias haya entre X e Y, mayor será su intersección y menor su distancia. En definitiva, más semejantes serán las preferencias de ambos jugadores.

II-B. Caso B: correlación entre preferencias cualitativas

Como en el caso anterior, los perfiles de los jugadores se expresan como conjuntos de características cualitativas. Pero si estas características no son independientes (p.e. enfermedades relacionadas) o no son binarias (p.e. expresadas como respuestas a cuestionarios en texto libre), la distancia entre los perfiles de dos jugadores no se puede calcular como el tamaño de la intersección entre sus conjuntos de preferencias. Por ejemplo, si $\mathcal{C}$ sufre anorexia y $\mathcal{S}$ bulimia, podemos determinar que existe cierta coincidencia entre ellos, puesto que ambos presentan desórdenes alimentarios. Esta coincidencia tiene que ser capturada por la distancia resultante.

Suponemos pues, que tenemos una función de correlación $s : E \times E \mapsto \mathbb{Z}_+$ que mide la semejanza entre los elementos de los conjuntos de características de $\mathcal{C}$ y $\mathcal{S}$, donde E es el dominio del que los conjuntos de características de ambos jugadores toman sus valores. Para características nominales (p.e. nombres de enfermedades), podemos utilizar semejanzas semánticas [13]; para características numéricas que tomen valores de dominios finitos y discretos (p.e. edades, códigos postales), podemos utilizar funciones aritméticas. Además, suponemos que ambos jugadores conocen s desde el principio.

En este caso la distancia entre el conjunto X de $\mathcal{C}$ y el conjunto Y de $\mathcal{S}$ se puede calcular como

$$1/(\textstyle\sum_{x \in X} \sum_{y \in Y} s(x, y))$$

cuando el denominador no es nulo. Si lo es, decimos que la distancia es ∞.

II-C. Caso C: funciones de preferencia cuantitativas

En este último caso, queremos calcular la diferencia entre dos funciones cuantitativas sobre el mismo dominio, que definen las preferencias o perfiles de los dos jugadores. Suponemos que éstas son funciones en $\mathbb{Z}$. Es decir, $\mathcal{C}$ tiene una función de preferencia privada $f : E \to \mathbb{Z}$ y $\mathcal{S}$ tiene una función privada $g : E \to \mathbb{Z}$.

Una manera de medir la distancia entre f y g es calcular $d(f, g) = \sum_{i=1}^{t} |f(x_i) - g(x_i)|$, donde $D = \{x_1, \ldots, x_t\}$ es un subconjunto discreto representativo de los elementos de E.

Este escenario encaja con las estrategias para aprender, modelar y gestionar perfiles de usuarios de redes sociales más habituales entre la literatura relacionada [1], [16], [21]. Éstas consisten normalmente en asociar un vector de pesos a cada usuario, donde cada peso expresa el interés de dicho usuario por cierto tema (p.e. deportes, ciencia, salud, etc.). Para comparar a dos usuarios, simplemente se calcula la distancia entre sus vectores de pesos.

III. Cálculo de distancias basado en la intersección de conjuntos

Más adelante mostraremos cómo los tres casos anteriores A, B y C se pueden reducir al cálculo del tamaño de la intersección de conjuntos. Por tanto, revisaremos primero los trabajos que proponen soluciones para el cálculo del tamaño de la intersección entre dos conjuntos de manera segura y privada. Para ello, nos centramos en ciertos protocolos de computación segura multiparte.

Los protocolos de computación segura multiparte (MPC, del inglés *secure multiparty computation*) permiten a un conjunto de entidades distintas calcular alguna función de sus entradas de manera segura y sin la necesidad de una entidad externa de confianza. Durante la ejecución del protocolo, las partes no descubren nada sobre los valores de entrada de los demás excepto todo aquello implicado por el resultado en sí. Principalmente, se tienen en cuenta dos modelos de adversarios: adversarios honestos-pero-curiosos y adversarios maliciosos. En el primero de los casos, las partes siguen las reglas del protocolo pero intentarán obtener información sobre los valores de entrada de las otras partes a partir de los mensajes que reciben. En el segundo, suponemos que el adversario se puede desviar del protocolo de cualquier modo.

Restringiremos nuestro caso a un protocolo en el que sólo participan dos partes, los valores de entrada de las cuales son

sendos conjuntos, y el resultado esperado es el cardinal de la intersección de éstos.

La intersección de dos conjuntos se puede obtener usando construcciones genéricas basadas en el protocolo de Yao [20]. Esta técnica permite calcular cualquier función aritmética, pero para la mayoría de funciones es ineficiente. Muchos de los trabajos recientes sobre protocolos de computación segura entre dos partes se centran en mejorar la eficiencia de estos protocolos para ciertas famílias de funciones.

Freedman, Nissim y Pinkas [4] presentaron un método más eficiente para calcular la intersección de conjuntos llamado *esquema de emparejamiento privado* (*private matching scheme*), que es seguro en el modelo honesto-pero-curioso. Un esquema de emparejamiento privado es un protocolo entre un cliente $\mathcal{C}$ y un servidor $\mathcal{S}$ en el que el valor de entrada de $\mathcal{C}$ es un conjunto X de tamaño $i_{\mathcal{C}}$, el valor de entrada de $\mathcal{S}$ es un conjunto Y de tamaño $i_{\mathcal{S}}$, y como resultado $\mathcal{C}$ obtiene $X \cap Y$. Este esquema usa técnicas basadas en polinomios y esquemas de cifrado homomórficos.

En [4] también se presentan algunas variaciones del esquema de emparejamiento privado: una extensión segura en presencia de adversarios maliciosos, una extensión para casos con más de dos partes y varias modificaciones para calcular el cardinal de la intersección y otras funciones. Construir esquemas eficientes para operaciones con conjuntos es un tema importante en MPC y ha sido estudiado en muchos otros trabajos. Diversas publicaciones, como [2], [3], [5], [10], [15], presentan nuevos protocolos para calcular el tamaño de la intersección de varios conjuntos.

A continuación especificaremos los protocolos para resolver los distintos casos, A, B y C, presentados anteriormente. En todos los casos, la distancia entre las preferencias privadas de las dos partes se calcula usando un protocolo de computación multiparte que devuelve el cardinal de la intersección de dos conjuntos.

III-A. Caso A

En este apartado describimos un protocolo en el que $\mathcal{C}$ aporta $X = \{a_1, \ldots, a_s\} \subseteq E$ y $\mathcal{S}$ aporta $Y = \{b_1, \ldots, b_t\} \subseteq E$, donde s y t son conocidos por ambos participantes. Finalmente, $\mathcal{C}$ obtiene $|X \cap Y|$. Para que $\mathcal{S}$ también obtenga $|X \cap Y|$, el protocolo debería ser ejecutado una segunda vez (de manera secuencial o concurrente) intercambiando los papeles de $\mathcal{C}$ y $\mathcal{S}$.

Usaremos el protocolo descrito en [4] para calcular el tamaño de la intersección de los conjuntos de entrada, que es seguro contra adversarios que siguen el modelo honesto-pero-curioso. Además, usaremos el criptosistema de Paillier [11] como esquema de cifrado homomórfico. El protocolo se aprovecha de la propiedad de este esquema que permite, dados tres elementos m_1, m_2, m_3, calcular eficientemente $Enc(m_1 + m_2)$ y $Enc(m_1 \cdot m_3)$ a partir de $Enc(m_1)$, $Enc(m_2)$, y m_3. Suponemos que $\mathcal{C}$ y $\mathcal{S}$ acuerdan una codificación común tanto para los elementos de E como para los elementos de la función Enc. Ambos acuerdan también una palabra especial m. Describimos el protocolo a continuación.

Paso 1. *$\mathcal{C}$ escoge los parámetros iniciales, genera su clave pública y privada, y publica tanto los parámetros como su clave pública.*

Paso 2. *$\mathcal{C}$ calcula, a partir de su conjunto X, el polinomio $p(x) = \prod_{i=1}^{s}(x - a_i)$.*

Paso 3. *$\mathcal{C}$ envía $Enc(p_0), \ldots, Enc(p_s)$ a $\mathcal{S}$, siendo cada p_i el coeficiente de grado i del polinomio p.*

Paso 4. *$\mathcal{S}$ genera los valores aleatorios $r_j \in \mathbb{Z}_n$ para todo $1 \leq j \leq t$. $\mathcal{S}$ calcula $Enc(r_j \cdot p(b_j) + m)$ para todo $1 \leq j \leq t$ y envía los textos cifrados a $\mathcal{C}$.*

Paso 5. *$\mathcal{C}$ descifra los t textos cifrados. El resultado de cada descifrado es m o un elemento aleatorio.*

Si el tamaño del dominio de Enc es mucho mayor que $|X|$ el esquema calcula $|X \cap Y|$ con una alta probabilidad: así, el número de mensajes m obtenidos en el último paso indica el número de elementos comunes en X e Y.

Nótese que $\mathcal{C}$ obtiene $|X \cap Y|$, pero no aprende ninguna información adicional sobre Y o $X \cap Y$ (en particular, $\mathcal{C}$ no puede determinar los elementos de estos conjuntos). Además, $\mathcal{S}$ no puede distinguir entre cada uno de los casos en que $\mathcal{C}$ proporcione conjuntos diferentes como entradas.

III-B. Caso B

En este caso, $\mathcal{C}$ proporciona X y $\mathcal{S}$ proporciona Y, dos conjuntos de características cualitativas, y quieren saber cuán cerca están estos conjuntos sin revelárselos al otro.

En el siguiente protocolo, sólo $\mathcal{C}$ obtiene la distancia entre X e Y; para que $\mathcal{S}$ también reciba esta información, el protocolo debería ser ejecutado de nuevo (de manera secuencial o concurrente) con los papeles de $\mathcal{C}$ y $\mathcal{S}$ intercambiados.

Supongamos que los dominios de X e Y son el mismo, llamémosle E. La cercanía o semejanza entre los elementos de E se calculará mediante una función s. En concreto, consideramos una función $s : E \times E \to \mathbb{Z}_+$. Nótese que el Caso A es un caso particular de este Caso B en el que $s(x, x) = 1$ y $s(x, y) = 0$ para todo $x \neq y$.

Sea Y la entrada de $\mathcal{S}$. Para todo $x \in E$, $\mathcal{S}$ calcula el valor $\ell_x = \sum_{y \in Y} s(x, y)$. Observemos que este valor ℓ_x mide la semejanza entre x e Y. Sea $Y' = \{y \in E : \ell_y > 0\}$. Es común considerar funciones que satisfagan $s(x, x) > 0$ para todo $x \in E$, por lo tanto en general $Y \subseteq Y'$.

Podemos obtener un protocolo para calcular tal función a partir del protocolo anterior, reemplazando el paso 4 por el siguiente:

Paso 4'. *$\mathcal{S}$ genera ℓ_y elementos aleatorios $r_1, \ldots, r_{\ell_y} \in \mathbb{Z}_n$. $\mathcal{S}$ calcula $Enc(r_j \cdot p(y) + m)$ para todo $1 \leq j \leq \ell_y$, y envía los términos cifrados a $\mathcal{C}$.*

Así, para todo $y \in Y'$, $\mathcal{S}$ envía ℓ_y textos cifrados. $\mathcal{C}$ recuperará m de ellos sólo si $y \in X$. Por lo tanto, al final del protocolo, el número total de mensajes descifrados que son iguales a m será

$$\sum_{x \in X} \ell_x = \sum_{x \in X} \sum_{y \in Y} s(x, y),$$

es decir, la suma de semejanzas entre los elementos de X e Y. Esto indica claramente cuán semejantes son X e Y. Del mismo modo que en el Caso A, ni $\mathcal{C}$ ni $\mathcal{S}$ obtienen ninguna información adicional de los elementos de los conjuntos de preferencias del otro.

III-C. Caso C

En este último caso, $\mathcal{C}$ introduce una función privada f y $\mathcal{S}$ una función privada g, y quieren medir la distancia d entre éstas sin revelarlas al otro.

El valor de $d(f,g)$ se calculará de manera vectorial. Suponemos que $f,g : E \to \mathbb{Z}_+$. Nótese que si f o g toman valores negativos, $\mathcal{C}$ y $\mathcal{S}$ pueden definir sendas funciones $f' : E \to \mathbb{Z}_+ : x \mapsto f(x) + c$ y $g' : E \to \mathbb{Z}_+ : x \mapsto g(x) + c$ para cierta constante $c \in \mathbb{Z}_+$ suficientemente grande. Obsérvese que $d(f,g) = d(f',g')$.

Dado el conjunto público $D = \{x_1, \cdots, x_t\} \subseteq E$, $\mathcal{C}$ define el vector $\mathbf{u} = (u_1, \ldots, u_t) \in \mathbb{Z}_+^t$, donde $u_i = f(x_i)$ for $i = 1, \ldots, t$, y $\mathcal{S}$ define $\mathbf{v} = (v_1, \ldots, v_t) \in \mathbb{Z}_+^t$, donde $v_i = g(x_i)$ for $i = 1, \ldots, t$. El problema descrito en la sección II-C se puede reducir a calcular $\|\mathbf{u} - \mathbf{v}\| = \sum_{i=1}^{t} |u_i - v_i|$.

Dados $\mathbf{u}$ y $\mathbf{v}$, definimos los conjuntos $X = \{(i,\ell) : u_i > 0 \text{ y } 1 \le \ell \le u_i\}$ e $Y = \{(i,\ell) : v_i > 0 \text{ y } 1 \le \ell \le v_i\}$. Siguiendo el protocolo para calcular el cardinal de la intersección de conjuntos presentado anteriormente, $\mathcal{C}$ y $\mathcal{S}$ pueden calcular $|X \cap Y|$ de manera privada (el protocolo se ha de ejecutar dos veces con los papeles de $\mathcal{C}$ y $\mathcal{S}$ intercambiados). Obsérvese que

$$\begin{aligned} |X \cap Y| &= |\{(i,\ell) : u_i, v_i > 0 \text{ y } 1 \le \ell \le \min\{u_i, v_i\}\}| \\ &= \sum_{1 \le i \le t} \min\{u_i, v_i\}. \end{aligned}$$

Según [4], además de obtener $|X \cap Y|$, durante el protocolo $\mathcal{S}$ aprende $|X|$ y $\mathcal{C}$ aprende $|Y|$. Por lo tanto $\mathcal{C}$ y $\mathcal{S}$ pueden calcular

$$\begin{aligned} &|X| + |Y| - 2|X \cap Y| = \\ &= \sum_{i=1}^{m} \max\{u_i, v_i\} + \min\{u_i, v_i\} - 2\sum_{i=1}^{m} \min\{u_i, v_i\} \\ &= \sum_{i=1}^{m} \max\{u_i, v_i\} - \min\{u_i, v_i\} \\ &= \sum_{i=1}^{m} |u_i - v_i| = \|\mathbf{u} - \mathbf{v}\| \end{aligned}$$

de manera privada.

IV. Análisis experimental

Esta sección ilustra la aplicabilidad de los protocolos propuestos para comparar perfiles de usuarios de redes sociales de manera que se preserve la privacidad entre ellos.

Hemos basado el experimento en 16 usuarios de Twitter seleccionados de entre los más relevantes en `WeFollow` [18] y `WhoToFollow` [19]. Estos sitios web ordenan y clasifican a los usuarios de Twitter en una serie de categorías. Tal como se hizo en [16], [17], tomamos a los dos usuarios más influyentes en 2012 dentro de cada una de las siguientes ocho categorías: Arte, Salud, Compras, Ciencia, Informática, Deportes, Sociedad y Negocios.

Tanto el cliente como el servidor se ejecutaron en el siguiente entorno: Asus S56C con Intel core i7 3517U, 8GB RAM DDR3 1600Mhz, Ubuntu 13.10 y Java7 (opendjk-1.7). La longitud de las claves es de 1024 bits. La implementación del criptosistema de Paillier que utilizamos es la proporcionada en [14], modificado para evaluar los polinomios usando el método de Horner.

Generamos un perfil para cada uno de los usuarios de Twitter siguiendo el proceso descrito en [16]. Resumiendo, extraemos las oraciones sustantivas de los últimos 100 tuits del usuario, y las clasificamos en las anteriores ocho categorías. Entonces, medimos la contribución de esa oración sustantiva a la categoría correspondiente como su capacidad informativa, calculada a partir de su distribución en la Web. Las contribuciones agregadas de todas las oraciones sustantivas de una categoría miden el interés del usuario en dicha categoría. Los perfiles son, por tanto, vectores normalizados que contienen ocho pesos, cada uno de ellos cuantificando el interés del usuario en cada una de las ocho categorías. Por ejemplo, el perfil de Twitter del usuario CERN, que corresponde al Centro Europeo de Investigación Nuclear, es {*Arte=15.1 %, Salud=0.27 %, Compras=1.79 %, Ciencia=47.93 %, Informática=7.5 %, Deportes=5.45 %, Sociedad=10.65 %, Negocios=11.31 %*}, lo que muestra una clara preferencia por temas relacionados con la ciencia. De este modo, los perfiles de usuario pueden ser entendidos como funciones de preferencia representables en un conjunto discreto de ocho elementos cuantitativos. Esto encaja con el Caso C y con el protocolo presentado en la sección III-C.

Para evaluar el comportamiento de nuestro protocolo en términos de precisión, primero calculamos las distancias d entre cada uno de los 16 perfiles tal y como describimos en la sección II-C: $d(f,g) = \sum_{i=1}^{m} |f(x_i) - g(x_i)|$, donde $x_i \in$ {*Arte, Salud, Compras, Ciencia, Informática, Deportes, Sociedad, Negocios*}, y f y g representan los perfiles de dos usuarios diferentes, asignando el peso de cada usuario para cada una de las categorías x_i. Luego, realizamos el mismo cálculo utilizando el protocolo descrito en la sección III-C.

Como nuestro protocolo supone que las funciones f y g tienen un dominio de valores enteros, en un primer momento redondeamos los pesos al entero más cercano. Para medir la precisión de los resultados, calculamos el error medio entre las distancias obtenidas sin preservar la privacidad y las obtenidas por nuestro protocolo. El error medio es 1.69 % con una desviación estándar del 2.25 %. Esto demuestra que nuestro protocolo no causa una distorsión significativa del resultado, más allá de la causada por el redondeo de las entradas.

Por otra parte, el tiempo medio de ejecución para calcular de manera privada la distancia entre dos perfiles es de 36.7 segundos, siendo inapreciable el tiempo en el caso normal. Analizando el protocolo (Sección III-C), podemos observar que el tiempo de ejecución depende del número de pesos que

comparemos (ocho) y de sus rangos. Ya que redondeamos los porcentajes para que sean enteros entre 0 y 100, el rango de pesos es de 100.

Si se diese un caso en el que el tiempo de respuesta fuese especialmente importante, podríamos sacrificar cierta precisión para acelerar el proceso, utilizando una representación de los pesos con un rango menor. Por ejemplo, si dividimos los pesos entre 10 y los redondeamos al entero más próximo, reducimos el rango de pesos a 10, lo que a cambio reduce el número de cifrados/descifrados en el protocolo en una orden de magnitud. Haciendo esto obtuvimos un tiempo medio de 2.7 segundos por cada uno de los cálculos de distancias. A cambio, el error medio respecto al cálculo normal fue del 18.49 % con una desviación típica del 17.8 %, lo que ilustra cómo la (falta de) precisión en la discretización de los valores de entrada afecta a la (falta de) precisión de la respuesta.

Por último, pero no menos importante, examinamos la escalabilidad del protocolo. La figura 1 muestra el incremento de los tiempos de ejecución para $\mathcal{C}$ y $\mathcal{S}$ cuando el tamaño de los conjuntos X e Y crece, teniendo $\mathcal{S}$ una carga computacional superior a $\mathcal{C}$. En la sección V discutimos la complejidad computacional de estos protocolos en mayor profundidad.

V. Trabajos relacionados

Consideramos el problema de calcular la distancia entre dos utilidades o funciones de preferencia privadas. En los casos descritos, tratamos dominios discretos o discretizados, lo que nos permite recurrir a la literatura sobre emparejamiento de registros privados. En este tipo de emparejamiento, el problema es ligeramente distinto: consiste en emparejar los registros de la misma entidad (un individuo, compañía, etc.) distribuidos en conjuntos de datos distintos, manteniendo la privacidad de estos registros. Existen principalmente tres estrategias para tratar este problema en la literatura: las basadas en la sanitización, las criptográficas y las híbridas.

En los métodos basados en sanitización, el emparejamiento se realiza sobre versiones perturbadas de los conjuntos de datos privados, con tal de protegerlos contra la revelación; en [6] se presenta un estudio de distintos métodos de perturbación/sanitización y emparejamiento de registros. Esta clase de métodos es generalmente eficiente, pero presenta ciertos problemas en la precisión: emparejar conjuntos de datos perturbados es evidentemente menos preciso que emparejar los originales. De hecho, se dan casos de falsos positivos y negativos.

Los métodos criptográficos se basan en MPC y proporcionan privacidad sin pérdida de precisión. Tal como hemos mencionado anteriormente, nosotros seguimos esta estrategia, ya que usamos MPC para calcular el tamaño de la intersección de conjuntos, específicamente el protocolo propuesto en [4]. Nuestra solución se podría adaptar fácilmente para utilizar otros protocolos para calcular lo mismo, como los que mencionamos al principio de la sección III ([2], [3], [5], [10], [15]).

La complejidad en las comunicaciones de nuestro protocolo es $O(i_{\mathcal{C}}+i_{\mathcal{S}})$, siendo $i_{\mathcal{C}}$ y $i_{\mathcal{S}}$ los tamaños de las entradas de $\mathcal{C}$ y $\mathcal{S}$, respectivamente. La complejidad computacional para $\mathcal{C}$ es $O(i_{\mathcal{C}}+i_{\mathcal{S}})$, mientras que la complejidad computacional para $\mathcal{S}$ es $O(i_{\mathcal{C}}i_{\mathcal{S}})$, pero se puede reducir hasta $O(i_{\mathcal{C}}\log\log i_{\mathcal{S}})$ [4]. La complejidad computacional del esquema presentado en [3] es lineal respecto a $i_{\mathcal{C}}+i_{\mathcal{S}}$. Otros protocolos, como los presentados en [10] no diferencian a los usuarios $\mathcal{C}$ y $\mathcal{S}$: ambos reciben el tamaño de la intersección al final del protocolo.

Existen también soluciones para calcular de manera privada el tamaño de la intersección de $n > 2$ conjuntos de n entidades, así como construcciones que son seguras frente al modelo de adversario malicioso [2], [3], [4], [5], [10], [15]. En un esquema de emparejamiento privado, los tamaños de las entradas son conocidos por ambas partes. Algunas técnicas presentadas en [2] permiten ocultar estos tamaños, a cambio de aumentar la complejidad en las comunicaciones y cálculos.

Los métodos híbridos intentan conseguir un equilibrio entre los métodos basados en sanitización y los métodos criptográficos, para mantener tanta precisión como sea posible sin aumentar drásticamente la complejidad computacional. La idea es introducir una fase en la que los conjuntos de datos se dividen en bloques, se sanitizan y se descartan aquellos bloques que no satisfacen las condiciones de emparejamiento. Tras esta fase, se aplican los métodos basados en MPC sobre el resto de bloques. Métodos como [7] (que utiliza k-anonimato), [8] (que utiliza privacidad diferencial) y [9] (que mejora [8]) siguen esta estrategia. La solución propuesta en este trabajo también se podría adaptar para seguir esta estrategia, ya que la fase en la que se usa MPC en esta estrategia se puede implementar como intersección de conjuntos.

VI. Conclusiones y trabajo futuro

Calcular la distancia entre las funciones de preferencia privadas de dos entidades es relevante en un amplio rango de aplicaciones. Hemos descrito varios escenarios de aplicación en los que funciones privadas expresan las preferencias o perfiles de las partes o, en términos de teoría de juegos, las utilidades de los jugadores. Estos escenarios incluyen encontrar amigos o socios con intereses parecidos en redes sociales, mitigar ataques de *grooming*, etc.

Hemos definido el problema para varios tipos de funciones privadas y, para cada una de ellas, hemos propuesto un protocolo que las resuelve basándonos en el cálculo seguro multiparte del tamaño de la intersección de conjuntos. El trabajo experimental muestra que preservar la privacidad de las preferencias no altera significativamente la precisión de las distancias obtenidas.

Siguiendo la misma línea de investigación, nos proponemos el diseño de protocolos eficientes para el cálculo de otras operaciones aritméticas que precisen de computación privada. Todo cálculo aritmético se puede realizar mediante un protocolo de computación multiparte [20], pero los métodos conocidos no son eficientes, en general. Además, consideramos posibles maneras de augmentar la precisión del cálculo, teniendo en cuenta todo el dominio E en lugar de un subconjunto D, sin

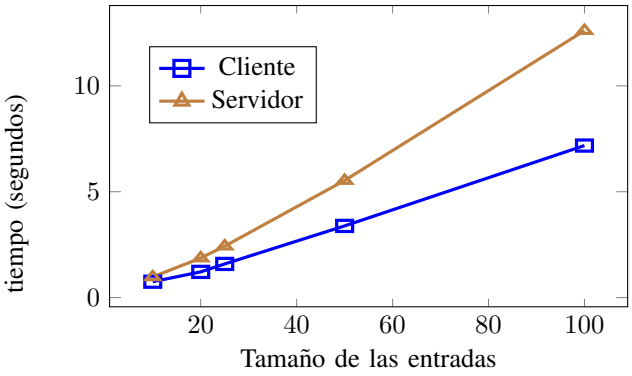


Figura 1. Tiempos de ejecución del cliente $\mathcal{C}$ y del servidor $\mathcal{S}$ para tamaños distintos.

que esto revierta en un augmento significativo de los costes computacionales.

AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por la Generalitat de Catalunya bajo la beca 2009 SGR 1135, por el Gobierno Español a través de los proyectos TIN2011-27076-C01-01 "CO-PRIVACY", TIN2012-32757 "ICWT", IPT-2012-0603-430000 "BallotNext" and CONSOLIDER INGENIO 2010 CSD2007-00004 "ARES", y por la Comisión Europea bajo los proyectos FP7 "DwB" e "Inter-Trust". J. Domingo-Ferrer está financiado parcialmente como investigador ICREA Acadèmia por la Generalitat de Catalunya. Los autores pertenecen a la Cátedra UNESCO de Privacidad de Datos, pero son los únicos responsables de las opiniones expresadas en este trabajo, que no reflejan necesariamente la posición de UNESCO ni comprometen a esta organización.

REFERENCIAS

[1] F. Abel, Q. Gao, G.J. Houben, K. Tao, "Semantic enrichment of Twitter posts for user profile construction on the social web", *ESWC'11*, pp. 375–389, 2011.
[2] M. Blanton, E. Aguiar, "Private and oblivious set and multiset operations", *ASIACCS 2012*, Springer, pp. 40–41, 2012.
[3] E. De Cristofaro, P. Gasti, G. Tsudik, "Fast and private computation of cardinality of set intersection and union", *CANS 2012*, Springer, pp. 218–231, 2012.
[4] J. Freedman, K. Nissim, B. Pinkas, "Efficient private matching and set intersection", *EUROCRYPT 2004*, Springer, pp. 1–19, 2004.
[5] S. Hohenberger, S. Weis, "Honest-verifier private disjointness testing without random oracles", *PET 2006*, Springer, pp. 277–294, 2006.
[6] A. Hundepool, J. Domingo-Ferrer, L. Franconi, S. Giessing, E. Schulte Nordholt, K. Spicer, P.-P. de Wolf, *Statistical Disclosure Control*, Wiley, 2012.
[7] A. Inan, M. Kantarcioglu, E. Bertino, M. Scannapieco, "A hybrid approach to private record linkage", *Proc. IEEE 24 Intl. Conf. Data Eng.-ICDE 2008*, pp. 496–505, 2008.
[8] A. Inan, M. Kantarcioglu, G. Ghinita, E. Bertino, "Private record matching using differential privacy", *Proc. of the 13th Intl. Conference on Extending Database Technology-EDBT'10*, pp. 123-134, 2010.
[9] A. Inan, M. Kantarcioglu, G. Ghinita, E. Bertino, "A hybrid approach to private record matching", *IEEE Transactions on Dependable and Secure Computing*, vol. 9(5), pp. 684–698, 2012.
[10] L. Kissner, D. X. Song, "Privacy-preserving set operations", *CRYPTO 2005*, Springer, pp. 241–257, 2005.
[11] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes", *EUROCRYPT 1999*, Springer, pp. 223–238, 1999.
[12] PatientsLikeMe, http://www.patientslikeme.com
[13] D. Sánchez, M. Batet, D. Isern, A. Valls, "Ontology-based semantic similarity: A new feature-based approach", *Expert Systems with Applications*, vol. 39(9), pp. 7718–7728, 2012.
[14] The Homomorphic Encryption Project (thep). https://code.google.com/p/thep/. Accessed 14 de julio de 2014.
[15] J. Vaidya, C. Clifton, "Secure set intersection cardinality with application to association rule mining", *Journal of Computer Security*, vol. 13(4), pp. 593–622, 2005.
[16] A. Viejo, D. Sánchez, J. Castellà-Roca, "Preventing automatic user profiling in Web 2.0 applications", *Knowledge-based Systems*, vol. 36, pp. 191–205, 2012.
[17] A. Viejo, D. Sánchez, J. Castellà-Roca, "Using profiling techniques to protect the user's privacy in Twitter", *MDAI 2012*, Springer, pp. 161–172, 2012.
[18] WeFollow, http://wefollow.com
[19] WhoToFollow, http://whotofollow.net
[20] A.C.-C. Yao, "How to generate and exchange secrets", *FOCS 1986*, pp. 162–167, 1986.
[21] K. Zoltan, S. Johann, "Semantic analysis of microposts for efficient people to people interactions", *RoEduNet11*, pp. 1–4, 2011.

Optimización en la generación de claves para firmas en anillo, espontáneas y enlazables

José Luis Salazar
Dpto. de Ingeniería Electrónica y Comunicaciones
Universidad de Zaragoza
Email: jsalazar@unizar.es

José Luis Tornos
Dpto. de Ingeniería Electrónica y Comunicaciones
Universidad de Zaragoza
Email: jltornos@unizar.es

***Resumen*—El gran desarrollo de las Tecnologías de la Información (TIC's) y su gran aceptación por parte de la ciudadanía ha permitido que múltiples sistemas de votación electrónica se desarrollen en los últimos años. Uno de los grandes retos es conseguir que estos sistemas puedan emplearse con una gran variedad de dispositivos, por lo que los protocolos implementados deberán poder ser empleados tanto en dispositivos con gran capacidad de cálculo y memoria (portátiles y ordenadores de sobre mesa principalmente) como por dispositivos con unas características más limitadas (tablets y smartphones). En este artículo se muestran dos estrategias distintas para el cálculo, desde el propio terminal móvil de votación, de las claves con las que los usuarios participarían en un proceso de votación electrónica y las características ofrecidas por cada una de ellas.**

***Palabras clave*—Generación de claves, Firmas en anillo, Votación electrónica (*Key generation, Ring Signatures, eVoting*).**

I. Introducción

El desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC), ha permitido implementar servicios en otros tiempos impensables como pueden ser la televisión a la carta o la telefonía móvil. Uno de estos servicios es la votación electrónica. Aunque podemos decir que todavía no existe una generalización del servicio (tampoco de la democracia), sí ha sido probado con éxito en varios países como son, entre otros, Suiza y Estonia [1], [2].

Por otra parte, con las nuevas arquitecturas de los terminales móviles la comunicación de datos se hace de una manera más liviana sin que la pila de protocolos de comunicación en Internet las constriña asfixiantemente. En el afán de simplificar todo lo posible el uso de la necesaria criptografía en las votaciones electrónicas, en este artículo proponemos un algoritmo de generación de claves para un sistema de votación portable en el que en una situación ideal, el votante pueda votar desde cualquier terminal móvil (propio o ajeno), sin una gran formación en TIC y con todas las garantías de seguridad telemática.

La restricción no es poca. Hay que tener en cuenta que de esta manera podemos dar acceso al votante a algún tipo de servicio, pero debemos dotar al terminal al que está conectado de todos las medidas de seguridad que requieren las comunicaciones implicadas en una votación electrónica y para ello dotarle (aunque sea de forma temporal) de capacidad de generación de las claves criptográficas necesarias. Para ello optamos por el modelo de una única entidad de confianza [3] que sea capaz de mantener el anonimato a través de una identificación previa.

Este modelo de una entidad de confianza basa el anonimato en el uso de firmas en anillo [4]. Además si queremos que el servicio sea portable, exigiremos que sean espontáneas; y para ser eficientes deberían tener una longitud fija (lo más corta posible). Luego una elección adecuada de firma para este tipo de votación electrónica es la propuesta por Wei [5]. Sin embargo, el aprovisionamiento de claves en este tipo de firmas no es trivial ya que las exigencias criptográficas hacen que el cálculo se haga desde el propio terminal móvil, obligándole a una carga computacional de la que no se está seguro se vaya a poder abastecer a priori.

En este artículo proponemos un modelo de generación de claves para este tipo de protocolo criptográfico que sea ejecutable en terminales con restricciones computacionales o de memoria, que pueda rellenar los campos de un Certificate Signing Request (CSR) de manera autónoma y almacenar la clave privada generada para su posterior uso en la votación electrónica. En el siguiente punto haremos una breve descripción del algoritmo de votación desarrollado a partir de las firmas en anillo y los requisitos que exigen sus claves. A continuación describiremos en profundidad nuestra propuesta para la generación de dichas claves, para finalmente exponer nuestros resultados y conclusiones.

II. Escenario

El escenario que planteamos [3] contempla tres actores diferenciados:

- El votante, que en principio no cuenta más que con un terminal para comunicarse y hacer las operaciones pertinentes.
- Una Autoridad de Certificacion (AC) que comprueba la identidad del votante y le emite un certificado asociado a un CSR enviado por éste. Por otra parte sirve de repositorio confiable de los parámetros asociados a un evento de votación, imprescindibles para construirse unas claves ad-hoc a dicha votación.
- La urna, encargada de recibir los votos, emitir justificantes de votos (si fuese necesario), recontarlos, hacer público los resultados y una lista de comprobación de votos (si también fuese necesario).

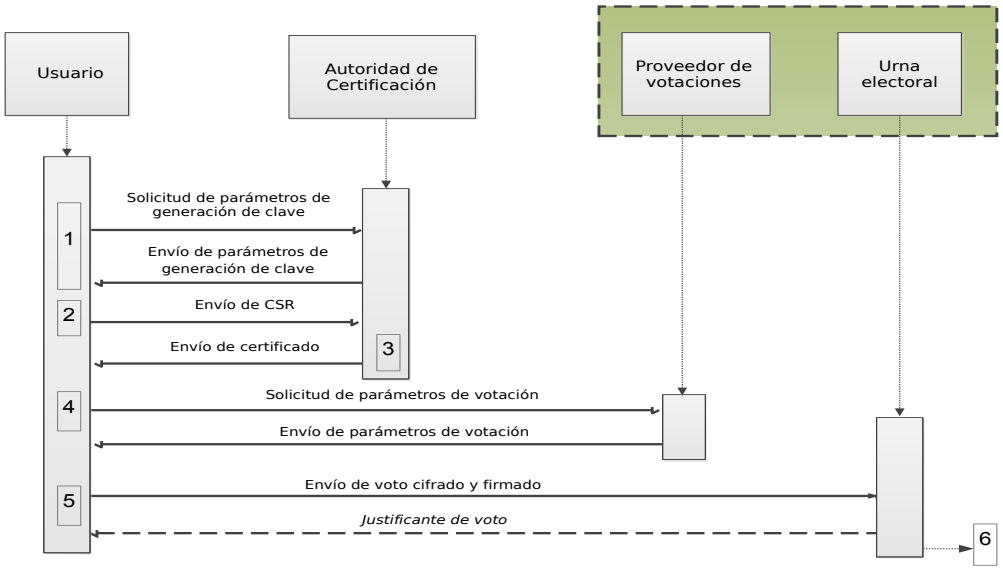


Figura 1. Proceso de votación

Contando con estos actores el proceso de votación es el siguiente (figura 1):

1. El votante se conecta con la Autoridad de Certificación y se descarga los parámetros necesarios (u, n, λ, μ) para generar las claves que empleará en una determinada votación.
2. El votante, desde su terminal, genera un par de claves (pública/privada), genera un CSR con su clave pública y sus datos identificativos y se lo envía, acompañado de unas credenciales de identidad previamente definidas, a la AC.
3. La AC verifica la identidad, genera el certificado de clave pública del usuario, y almacena el certificado (registro del votante) ya que será necesario para el cálculo de uno de los parámetros de la votación. Finalmente envía el certificado de clave pública al usuario, que lo almacenará en formato PKCS12 junto con su clave privada.
4. Una vez comenzado el periodo de votación, y con los parámetros creados por la urna, el votante los descarga. Entre ellos está el conjunto de las claves públicas de todos los participantes en la votación.
5. Una vez obtenidos los parámetros, cada usuario vota, cifrando el voto con la clave pública de la urna y firmándolo en anillo con las claves asociadas al certificado emitido para él. En caso de ser necesario, recibe de la urna el justificante del voto.
6. La urna recuenta los votos y los hace públicos (en caso de no emitir recibos) o hace públicos los resultados y las etiquetas asociadas a los votantes (sin mostrar relación alguna entre ellos).

II-A. Parámetros para la generación de claves

Una vez establecido cuál es el escenario de votaciones, veamos cuáles son los parámetros necesarios para la creación de las claves asociadas a una votación determinada. Es decir, para la implementación del paso 2 a partir de los parámetros que el proveedor de votaciones envía al usuario en el paso 1.

Según lo establecido en [5] el usuario ha obtenido de la AC los parámetros asociados a una votación: u, n, λ y μ. El parámetro λ mide la seguridad del sistema a través del tamaño de n, siendo $n = pq = (2p' + 1)(2q' + 1)$, con p, q, p', q' primos y $p', q' > 2^{\frac{\lambda}{2}}$, siendo u, un residuo cuadrático no trivial modulo n. Además $|e_1 - 2^l|, |e_2 - 2^l| < 2^\mu$, donde λ y μ se eligen para evitar ataques de coalición [6] y bastaría con tomar $l \approx \lambda/2$ y μ lo suficientemente pequeño como se establece en [5] para evitarlos. En nuestro caso tomamos $\mu = l - 2$ siguiendo la recomendación de [9].

Si bien la generación de n, a través de p, q, p' y q' no es inmediata, se trata de un caso específico de generación de primos robustos [7], que está bastante estandarizado y queda fuera del alcance de este artículo.

II-B. Generación de claves de usuario

Llamaremos al algoritmo de generación de claves de usuario Key_gen(u, n, λ, μ) = $\{e_1, e_2, x = 2e_1e_2 + 1\}$, siendo los tres valores e_1, e_2 y x primos. Cada par e_1, e_2 conforma una clave privada para la firma en anillo, siendo x la clave pública asociada a dichos valores.

Con esta restricción se puede comprobar fácilmente que $0 \neq e_1 \neq e_2 \neq 0 \pmod 3$. Sin pérdida de generalidad tenemos que $e_1 = 1 \pmod 3$ y $e_2 = 2 \pmod 3$. Por lo tanto $x = 2 \pmod 3$, y $e_1, e_2 \in (2^{\frac{\lambda}{2}} - 2^\mu, 2^{\frac{\lambda}{2}} + 2^\mu)$. La estrategia para la búsqueda de estas claves estará condicionada por la

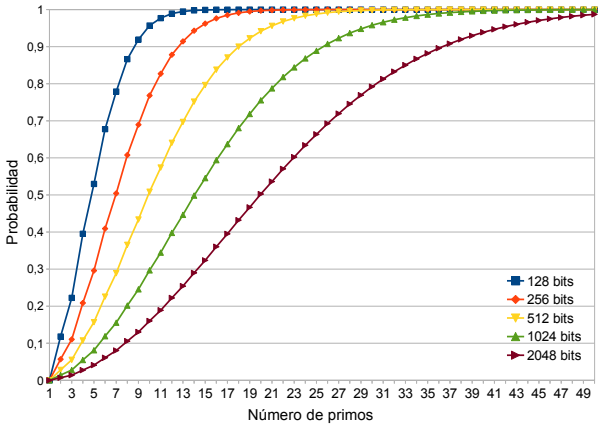


Figura 2. Número de primos para obtener un primo $2e_1e_2+1$

capacidad computacional, tanto de tiempo de procesado, como de almacenamiento en memoria.

Sin embargo, en cualquiera de los casos, la búsqueda ha de comenzar a partir de la generación de los dos primos pequeños, para posteriormente generar el grande. Por lo tanto, podemos suponer que el tiempo de comprobación de la primalidad de un número $y \in (2^{\frac{\lambda}{2}}-2^{\mu}, 2^{\frac{\lambda}{2}}+2^{\mu})$ es prácticamente constante, dependiente de λ, y que el número de comprobaciones necesarias hasta encontrar dicho número primo viene determinada por la distribución de los números primos en el intervalo. Esa distribución nos dará una función de la probabilidad de que al elegir un número entero aleatorio con distribución uniforme en ese intervalo, dicho número sea primo y lo denotaremos con $p_{\lambda,\mu}$.

Por otro lado una vez determinado que y es primo sabemos que P(y = 1 (mod 3)) = 1/2 = P(y = 2 (mod 3)), dada la distribución de los números primos en sucesiones aritméticas. Con esto y teniendo en cuenta el teorema del número primo, aproximaremos la esperanza de intentos para encontrar un primo y como ($1/p_{\lambda,\mu}$) y la de que tenga una determinada congruencia con 1 ó 2 módulo 3 será el doble de dicho valor, ($2/p_{\lambda,\mu}$).

Antes de comenzar nuestro análisis daremos una cota superior de una probabilidad que nos será muy útil. Sería interesante conocer cuál es la probabilidad p de que a partir de un número e_1 dado, encontrar otro número primo $x = 2ke_1+1$, con k variando en el intervalo ($2^{\frac{\lambda}{2}}$-2^{μ}, $2^{\frac{\lambda}{2}}$+2^{μ}), con lo que $x \in (1+2^{\lambda+1}+2^{2\mu+1}-2^{\frac{\lambda}{2}+\mu+2}, 1+2^{\lambda+1}+2^{2\mu+1}+2^{\frac{\lambda}{2}+\mu+2})$.

Utilizando la notación del teorema de Brun-Titchmarsh [8], la expresión $\prod(a,b,c)$ denota la cantidad de números primos menores que a en la sucesión $x_n = c+nb$. De este teorema podemos deducir que la densidad de números primos en esa sucesión, dentro del intervalo ($2^{\frac{\lambda}{2}}$-2^{μ}, $2^{\frac{\lambda}{2}}$+2^{μ}), va a estar acotada por:

$$\prod(1+2^{\lambda+1}+2^{2\mu+1}+2^{\frac{\lambda}{2}+\mu+2},2e_1,1)-\prod(1+2^{\lambda+1}+2^{2\mu+1}-2^{\frac{\lambda}{2}+\mu+2},2e_1,1) < \frac{2\cdot 2^{\frac{\lambda}{2}+\mu+3}}{\varphi(2e_1)log(\frac{2^{\frac{\lambda}{2}+\mu+3}}{2e_1})} = \frac{2^{\frac{\lambda}{2}+\mu+4}}{(e_1-1)log(\frac{2^{\frac{\lambda}{2}+\mu+2}}{e_1})};$$

Por lo tanto, la probabilidad buscada p estará acotada por: $p = \frac{\prod(1+2^{\lambda+1}+2^{2\mu+1}+2^{\frac{\lambda}{2}+\mu+2},2e_1,1)-\prod(1+2^{\lambda+1}+2^{2\mu+1}-2^{\frac{\lambda}{2}+\mu+2},2e_1,1)}{2^{\frac{\lambda}{2}+\mu+3}/2e_1}$

$$< \frac{\frac{2^{\frac{\lambda}{2}+\mu+4}}{(e_1-1)log\left(\frac{2^{\frac{\lambda}{2}+\mu+2}}{e_1}\right)}}{\frac{2^{\frac{\lambda}{2}+\mu+2}}{e_1}} = \frac{4e_1}{(e_1-1)log\left(\frac{2^{\frac{\lambda}{2}+\mu+2}}{e_1}\right)} \approx \frac{4}{log\left(\frac{2^{\frac{\lambda}{2}+\mu+2}}{e_1}\right)} \leq \frac{4}{log\left(\frac{2^{\frac{\lambda}{2}+\mu+2}}{2^{\frac{\lambda}{2}+2^{\mu}}}\right)}$$

III. Estrategias de minimización

III-A. *Estrategias de minimización de memoria*

En una estrategia de minimización de memoria, deberíamos fijar e_i con $i \in \{1,2\}$ que lo conseguiremos tras un número medio de intentos ($1/p_{\lambda,\mu}$). Ahora deberíamos buscar primos e_j con $i \neq j \in \{1,2\}$ lo cual conseguiremos tras ($2/p_{\lambda,\mu}$). Si $x = 2e_1e_2+1$ es primo, ya habríamos conseguido nuestro objetivo, en caso contrario buscaríamos otro e_j. En resumidas cuentas, estaríamos buscando números primos en la sucesión aritmética $x_{ik} = 1+2e_ik$. Aunque el número k ha de ser primo y por lo tanto su distribución deja de ser uniforme, consideraremos (perdiendo una mínima precisión) que sí lo es ya que previamente hemos calculado la cota de probabilidad p sobre una distribución uniforme (cota que también será válida para esta otra distribución). En ese caso, el número de $k's$ necesarias, será mayor que la esperanza de la distribución geométrica con probabilidad de éxito, p, es decir $1/p$. El algoritmo sería el siguiente:

Key_gen(u, n, λ, μ):

1. $e_i = Rand(\lambda,\mu)$
2. WHILE (e_i no primo) $\{e_i = Rand(\lambda,\mu)\}$
3. $e_j = 1$, $x = 2e_ie_j$
4. WHILE (x no primo)
   $\{e_j = Rand(\lambda,\mu)$
   WHILE (e_j no primo) $\{e_j = Rand(\lambda,\mu)\}$
   $x = 2e_1e_2+1\}$
5. RETURN (e_1,e_2,x)

Teniendo en cuenta que para el cálculo del tiempo de procesado, la tarea que más tiempo requiere es calcular la primalidad de x, podemos afirmar que el tiempo de procesado es del orden de $1/p$. La cantidad de memoria de almacenamiento empleada se mantiene constante a lo largo de todo el proceso y se limita a dos únicos primos, el primo e_i que permanece fijo y los $e_j's, i \neq j$, necesarios hasta obtener un primo de la forma $2e_je_i+1$.

III-B. *Estrategias de minimización de tiempo de procesado*

Si por el contrario lo que queremos es seguir una estrategia de optimización del tiempo de procesado, tendremos que minimizar el número de test de primalidad para números del tamaño de x. Analizando nuestro objetivo podemos apreciar que el número sobre el que queremos aplicar los test

de primalidad tiene en realidad dos fuentes de aleatoriedad prácticamente independientes e_1 y e_2, por lo que quizás sería conveniente pensar en intentar usar la paradoja del cumpleaños en nuestro beneficio.

Para ello definiremos inicialmente 2 conjuntos: P_1 y P_2, donde $P_i = \{x \in (2^{\frac{\lambda}{2}} - 2^{\mu}, 2^{\frac{\lambda}{2}} + 2^{\mu}) \backslash x$ es primo y $x = i(mod\ 3)\}$. Se generan números primos en el intervalo $(2^{\frac{\lambda}{2}} - 2^{\mu}, 2^{\frac{\lambda}{2}} + 2^{\mu})$ con una semilla aleatoria con distribución uniforme. Cada vez que consigamos un primo, r, veremos a qué conjunto P_i pertenece e iremos comprobando la primalidad de $x = 2re_j+1$, donde e_j va recorriendo todos los valores de P_j, $(j = -i(mod\ 3))$. Si no se consigue ningún resultado positivo, añadimos r al conjunto P_i e iniciamos la búsqueda de un nuevo r. El algoritmo sería el siguiente:
Key_gen(u, n, λ, μ):

1. $P_1 = \emptyset$, $P_2 = \emptyset$; $i_1 = 0, i_2 = 0$
2. WHILE ($P_1 = \emptyset$ OR $P_2 = \emptyset$)
   $\{r = Rand(\lambda, \mu)$
   WHILE (r no primo) $\{r = Rand(\lambda, \mu)\}$
   $k = r\ mod(3)$
   $P_k[i_k] = r$
   $i_k + +\}$
3. Éxito = False; $i = 0$
4. WHILE (!Éxito AND $i < \#P_{-k}$)
   $\{x = 1 + 2rP_{-k}[i]$
   IF (x primo) {Éxito =TRUE}
   $i + +\ \}$
5. WHILE (!Éxito)
   $\{r = Rand(\lambda, \mu)$
   WHILE (r no primo) $\{r = Rand(\lambda, \mu)\}$
   $k = r\ mod(3)$
   $i = 0$
   WHILE (!Éxito AND $i < \#P_{-k}$)
   $\{x = 1 + 2rP_{-k}[i]$
   IF (x primo) {Éxito =TRUE}
   IF (!Éxito) {i++}}
   IF (!Éxito) $\{P_k[i] = r\}\ \}$
6. RETURN $(r, P_{-k}[i], x)$

En este caso si consideramos que P_1 y P_2 están permanentemente equilibrados y llamamos k al número de $r's$ calculadas obtendremos que para un entero m:
$P(k \leq m) = 1 - P(k > m) = 1 - q^{\left(\frac{m-1}{2}\right)^2}$, donde hemos definido q = 1 - p.

Calcular la esperanza analítica de esta distribución, y por tanto de la memoria necesaria, no resulta sencillo. La memoria necesaria aumentará conforme más primos e_1 y e_2 sean necesarios para el cálculo del primo $2e_1e_2 + 1$. A priori no se conoce el total de memoria necesaria pero sí que se puede realizar una cierta estimación basándonos en los resultados teóricos obtenidos, figura 2. En la Sección IV hacemos una comparativa sobre las gráficas teóricas resultantes de los valores de p para diferentes valores de λ y exponemos los resultados empíricos para el cálculo de claves reales con las características exigidas.

IV. Resultados

En la figura 2 podemos apreciar cuál sería la curva teórica para la distribución de la probabilidad del número de intentos necesarios para el cálculo de claves en el caso de minimización de tiempo de procesado. Podemos ver cómo el número de primos que debemos extraer será mayor conforme aumente la longitud de las claves que queremos calcular.

En la tabla I indicamos los resultados obtenidos después de realizar el cálculo de 500 claves de cada tipo: número medio de primos, e_1 y e_2 calculados; el número de intentos que se han realizado antes de obtener el primo buscado; y el tiempo que se ha tardado en obtenerlo. En las dos últimas columnas indicamos el valor teórico con el que se supera el umbral del 0.5 de probabilidad de obtener un primo $2e_1e_2 + 1$ y su probabilidad asociada. Los valores de $\#P_1$ y $\#P_2$ no están perfectamente equilibrados pero la mayor diferencia entre las medias de e_1 y e_2 calculados es menor del 3 %.

En la tabla II se muestran los resultados para la estrategia de optimización de memoria. En este caso se muestran los valores de la esperanza de este sistema para obtener las claves, $(1/p_{\lambda\mu})$. Si comparamos el valor del número de extracciones esperadas para la obtención de un primo $2e_1e_2+1$, vemos que la diferencia con la estrategia de minimización de tiempo es más que considerable y además esta diferencia se incrementa conforme mayor es el tamaño de la clave buscada, llegando a ser la esperanza del número de intentos de esta estrategia mas de siete veces mayor que en la estrategia de minimización del tiempo de procesado.

V. Conclusiones

En el artículo se muestra el proceso de generación de claves para un protocolo de votación electrónica. Se han mostrado dos estrategias distintas de obtención de claves, minimización de memoria y minimización de tiempo de procesado, y se ha

Tabla I
Resultados con minimización de tiempo

Bits	$\#P_1$	$\#P_2$	**# x fallidas**	**Tiempo(ms)**	m	**P(m)**
128	5,55	5,62	33	5,5	5	0,52976
256	7,70	7,48	65	30,5	7	0,50432
512	10,40	10,64	126	262,6	10	0,50875
1024	14,21	14,64	251	3319,9	15	0,54542
2048	21,23	21,37	544	56664,1	20	0,50248

Tabla II
Probabilidad y esperanza para minimización de memoria

Bits	**E[nº intentos]**
128	8,46254
256	17,6031
512	35,6735
1024	71,5329
2048	143,744

hecho un desarrollo teórico del coste de obtención de dichas claves con las dos estrategias.

Podemos concluir que ambas estrategias son válidas para el efecto que se diseñaron. Sin embargo, podemos apreciar que pensar en una estrategia de minimización de memoria exige una cantidad y tiempo de procesado que es difícil de asumir en los dispositivos actuales ya que su coste (ya sea tiempo o precio del procesador) parece resultar más caro que la memoria necesaria para acelerar el proceso.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto CPUFLIPI (MICINN TIN2010-17298) del Gobierno de España y por la Cátedra Telefónica-Universidad de Zaragoza.

REFERENCIAS

[1] J. Gerlach, U. Gasser, "Three Case Studies from Switzerland: E-Voting" *Berkman Center Research Publication No. 2009-03.1*

[2] U. Madise, P. Vinkel, "Constitutionality of Remote Internet Voting: The Estonian Perspective," *Juridica International XVIII*. 2011, pp. 4-16

[3] J.L. Salazar, J. Piles, J. Ruiz, J.M. Moreno-Jiménez, "Security approaches in e-cognocracy," *Computer Standards and Interfaces, vol. 32*, 2010 pp. 256–265.

[4] R.L. Rivest, A. Shamir and Y. Tauman, "How to leak a secret," *Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '01)*, 2001, pp. 552–565.

[5] P. P. Tsang, V. K. Wei, "Short linkable ring signatures for e-voting, e-cash and attestation," *Proceedings of the First international conference on Information Security Practice and Experience (ISPEC'05)*, 2005, pp. 48–60.

[6] Camenisch, J., Lysyanskaya, A., "A signature scheme with efficient protocols," Cimato, S., Galdi, C., Persiano, G. (eds.)*Security Communication Networks, vol. 2576*, 2002, pp. 268-289

[7] J. Gordon, "Strong Primes are Easy to Find," *Advances in Criptology - Eurocrypt'84.*, 1984, pp. 216–223

[8] J. Friedlander, H. Iwaniec, 'Applications to Linear Sequences (57)," American Mathematical Society Colloquium Publications. American Mathematical Society, Providence, RI, 2010.

[9] G. Ateniese, J. Camenischm M. Joye, G. Tsudik "A Practical and Provably Secure Coalition-Resistant Group Signature Scheme," CRYPTO 2000. LNCS, vol. 1880, pp. 255–270. Springer, Heidelberg (2000)

Un Enfoque Tolerante a Interrupciones para la Seguridad de la Internet de las Cosas

Daniel Ezquerra, Àngela Fabregas, MCarmen de Toro, Joan Borrell
Depto. de Ingeniería de la Información y de las Comunicaciones
Universitat Autònoma de Barcelona - Escola d'Enginyeria - Edifici Q
08193 Bellaterra, Spain
Email: {daniel.ezquerra,angela.fabregues,mcarmen.detoro,joan.borrell}@deic.uab.cat

***Resumen*—La Internet de las cosas (IoT, *Internet of Things*) es un paradigma emergente que pretende la interconexión de cualquier objeto susceptible de contar con una parte de electrónica, favorecido por la miniaturización de los componentes. El estado de desarrollo de la IoT hace que no haya ninguna propuesta firme para garantizar la seguridad y la comunicación extremo a extremo. En este artículo presentamos un trabajo en progreso hacia una aproximación tolerante a retrasos (DTN, *Delay and Disruption Tolerant Networks*) para la comunicación en el paradigma de la IoT y planteamos la adaptación de los mecanismo de seguridad existentes en DTN a la IoT.**

***Palabras clave*—Internet of Things(IoT), Redes tolerantes a retrasos e interrupciones(DTN), Seguridad en IoT.**

I. Introducción

La evolución de la tecnología ha permitido la miniaturización de muchos componentes electrónicos y junto con esta miniaturización ha aparecido la posibilidad de interconectar millones de dispositivos. Mientras que hasta ahora la mayoría de dispositivos interconectados eran controlados por humanos, esta evolución permitirá la comunicación máquina a máquina u objeto a objeto, con el fin de cooperar y lograr objetivos comunes. Es esta la idea que define la Internet de las cosas (IoT, *Internet of Things*) [1] y la Internet de las nano-cosas (IoNT, *Internet of Nano-Things*) [2], nuevos paradigmas en el escenario de las redes no cableadas.

Si consideramos la comunicación entre cualquier tipo de objeto, más allá de la comunicación entre ordenadores personales o teléfonos inteligentes, encontramos que el número de objetos conectados podría superar los 100 billones [3]. Si además consideramos los nano dispositivos podemos encontrarnos en un escenario donde se podría producir una monitorización constante de datos sensibles, como datos referentes a la salud (constantes vitales de un usuario) o las posiciones geográficas. Esta es la razón por la cual, antes de que la IoT pueda ser ampliamente aceptada, es el momento de trabajar en la seguridad de este paradigma.

Existe un gran número de propuestas que intentan definir los límites de lo que debe ser la IoT y cuales deben ser los bloques que la conformen. Asimismo se trata de definir los estándares y las visiones que serán ampliamente aceptadas en este nuevo paradigma. Igual pasa en otros paradigmas emergentes, como es el caso de la arquitectura de red tolerante a retrasos (DTN, *Delay and Disruption Tolerant Networks*)[4], dónde están apareciendo trabajos que tratan de dar respuesta a necesidades de seguridad y comunicación.

En este artículo presentamos un trabajo en progreso hacia una aproximación a la seguridad de la IoT mediante un un enfoque tolerante a interrupciones y retrasos basado en DTN.

La arquitectura DTN está pensada para escenarios con grandes retrasos, donde no es posible la comunicación extremo a extremo y donde existen una gran variedad de dispositivos. Debido a su diseño, la DTN es una arquitectura que podría resultar adecuada para dar respuesta a las necesidades comunicativas de la IoT. Al mismo tiempo, diversas soluciones de seguridad han sido estudiadas para cubrir las necesidades de la arquitectura DTN. Debido a la similitud entre las necesidades de las DTN y las necesidades de la IoT, creemos que las medidas de seguridad con validez en el campo de las DTN podrían ser adaptadas para usarse en la IoT.

El resto del artículo está estructurado de la siguiente manera: en la segunda sección presentamos el trabajo previo relacionado. En la tercera sección planteamos una adecuación de DTN a la IoT. En la cuarta sección presentamos nuestro trabajo en progreso sobre la seguridad en la IoT. En la quinta sección describimos un escenario de aplicación para la aproximación propuesta. Las conclusiones y las líneas de trabajo futuro cierran el artículo.

II. Seguridad en la IoT

En los trabajos concernientes a la IoT existen dos grandes aproximaciones para convertir el paradigma planteado en una realidad: aproximaciones centralizadas y distribuidas. En las siguientes secciones presentamos la seguridad según los condicionantes de cada una de las visiones y otras propuestas independientes a la aproximación planteada.

II-A. Aspectos de seguridad inherentes al diseño

La IoT [1] es posible gracias a diversas tecnologías que agrupan desde sistemas pasivos de identificación como RFID [5], hasta sistemas más complejos en los que objetos y dispositivos son capaces de generar datos y comunicarse con otros objetos sin necesidad de intervención externa. Las redes de sensores [6] o las redes móviles ad-hoc [7] son otras de las tecnologías que conforman la IoT, juntamente con las redes intracorporales propuestas dentro del ámbito de IoNT [2].

La mayoría de aproximaciones ofrecidas para la IoT que se encuentran en funcionamiento acostumbran a tener un diseño centralizado, en el que varios dispositivos identificadores, como podrían ser marcadores RFID o sensores, envían la información a un servidor central, controlado por el proveedor del servicio. La ventaja de estos sistemas es que el fabricante puede ofrecer soluciones de criptografía simétrica entre los dispositivos y el servidor, y posteriormente securizar el servidor con los mecanismos que ofrece una red como Internet en la que existe comunicación de extremo a extremo. Muchas de las propuestas de seguridad concernientes a RFID se basan en este concepto [8], [9].

Otros trabajos, como el presentado en [10], estudian la posibilidad de una IoT distribuida o híbrida en lugar de un esquema completamente centralizado como el mencionado. El esquema distribuido o híbrido supone ciertas desventajas en la aplicación de medidas de seguridad, puesto que se complica la implementación de mecanismos conocidos, como podrían ser sistemas de autenticación basados en una infraestructura de clave pública (PKI, *Private Key Infarestructure*). Por el contrario, el planteamiento de una arquitectura distribuida permite estar más cerca de un paradigma en el que los dispositivos se comunican entre ellos para lograr objetivos comunes o cooperativos en el ámbito local. Un sistema distribuido permite, además, una mayor escalabilidad y, al no estar concentrada toda la inteligencia en un solo dispositivo, posibilita la implementación de políticas de privacidad u otros tratamientos de datos más allá de guardar o recuperar información.

Independientemente del planteamiento distribuido o centralizado, se están realizando esfuerzos para crear estándares necesarios para ofrecer protocolos de comunicación y seguridad adecuados. Uno de los protocolos más usados para conseguir la comunicación entre elementos con restricciones de recursos es ZigBee [11] junto con una adaptación del protocolo IPv6 conocida como LowPan6 [12]. La conjunción de ambos permite trabajar con redes de sensores y redes ad-hoc de manera que cada elemento pueda tener un identificador.

II-B. Soluciones de seguridad

Las propuestas de seguridad especificas para redes ad-hoc móviles, redes inalámbricas malladas [13] o redes de sensores pueden ser buenas aproximaciones para la seguridad en la IoT dado que este tipo de redes se presentan como parte de los bloques que constituyen el paradigma de interconexión de objetos. Por las características de estas redes, los mecanismos criptográficos basados en PKI suponen un sobrecoste que no siempre resulta asumible. Este sobrecoste, añadido al hecho que en las redes de sensores no siempre es posible contactar con una autoridad certificadora, hace que la criptografía basada en PKI no sea siempre una buena solución.

ZigBee aporta, también, sus propios mecanismos de seguridad, aunque están diseñados para redes del tipo muchos a uno. Es decir, no está pensado para una arquitectura distribuida, en la que podría presentar problemas de escalabilidad. Además en [14] se realiza un estudio de varios mecanismos de seguridad para redes inalámbricas malladas (WMN, *Wireless Mesh Networks*) en los que quedan patentes algunos de los problemas de ZigBee respecto a la confidencialidad. Asimismo, el uso de entidades coordinadoras dificultan la gestión de claves.

Algunos estudios, como los presentados en [15], hacen uso de criptografía basada en pairings (PBC *Pairing Based Criptography*) tratando de solventar los problema de gestión de claves que plantean otros sistemas criptográficos. Aún con propuestas que prescinden de usar una PKI, como aquellas basadas en pairings, la distribución de claves en un entorno como el de la IoT, con un gran número de dispositivos conectados, plantea problemas de escalabilidad. Tratando de solventar los problemas de distribución de claves, se encuentran algunas propuestas que ofrecen soluciones mediante criptografía basada en la identidad [16] (IBC, *Identity Based Criptography*).

Los problemas de integridad y confidencialidad, sin embargo, no son los únicos problemas de seguridad que se encuentran en la IoT. Un entorno en el que existen grandes cantidades de objetos interconectados supone un problema para la privacidad y el anonimato de los usuarios. Deben diseñarse sistemas de autenticación que limiten de forma eficiente quien puede recuperar los datos de sensores u otros objetos. En este aspecto, existen estudios como [17] que proponen soluciones a los problemas de autenticación. Otras aproximaciones como la planteada en [18], no hacen uso del citado tipo de criptografía. Sin embargo la propuesta se ha demostrado insegura en [19].

III. Aproximación DTN a la IoT

En esta sección describimos la arquitectura DTN y justificamos la aproximación DTN para la IoT a la vez que revisamos las limitaciones de nuestra aproximación.

III-A. Redes tolerantes a retraos e interrupciones

DTN [4] es una arquitectura de red diseñada para trabajar en entornos sin conectividad extremo a extremo, con grandes retrasos en la comunicación, canales asimétricos y dispositivos heterogéneos. DTN hace uso de mecanismos de *store-carry and forward* que permiten a un nodo almacenar los mensajes mientras no hay comunicación y entregarlos en cuando se produce el contacto con otros nodos. La comunicación de este tipo de redes es, en ocasiones, de tipo oportunista y está supeditada al encuentro con otros nodos. Para garantizar la comunicación extremo a extremo se define el protocolo Bundle [20] que permite la entrega de los mensajes sin importar los protocolos subyacentes. Algunos trabajos presentan enfoques DTN para entornos en los que son aplicables otras arquitecturas, como en [21] para redes de sensores o [22] para redes malladas.

III-B. Aproximación DTN a la IoT

El paradigma IoT se construye a partir de un conjunto de tecnologías heterogéneas que comparten unas restricciones comunes como una capacidad de cómputo limitada y limitaciones energéticas. Sin embargo, no todos los dispositivos en IoT sufren de estas restricciones puesto que existen dispositivos

que poseen mayor capacidad de cálculo y que pueden tener un acceso constante a una fuente de energía.

Dado lo heterogéneo de los dispositivos que pueden conformar IoT, en la aproximación que proponemos se entenderá una región como un conjunto de dispositivos que comparten unas características similares en capacidad de computo, limitaciones de batería y capacidad de interconexión.

Teniendo en cuenta los distintos tipos de dispositivos, la aproximación DTN podría considerar una arquitectura de comunicación híbrida, en la que existen regiones de comunicación muchos a uno, como en el caso de RFID, o muchos a muchos, como el caso de algunas redes de sensores, que además se comunican entre ellas. Así pues, con una arquitectura híbrida los nodos de la DTN equivaldrían a los sensores u objetos capaces de identificarse, como los marcadores RFID. Por otra parte, dispositivos con mayor capacidad de cómputo y batería podrían actuar como mulas de datos, es decir, podrían recoger los datos de otros nodos para reenviarlos. El uso del protocolo Bundle [20] garantizaría las comunicaciones extremo a extremo y su extensión de seguridad permitiría otorgar confidencialidad e integridad a los datos [23]. La finalidad de usar una arquitectura DTN sería proveer comunicación entre las regiones y entre los dispositivos de la misma región. Las características de diseño de la arquitectura DTN complementarían a la IoT supliendo algunas de las restricciones necesarias como falta de disponibilidad, canales asimétricos o el trabajo con dispositivos heterogéneos.

III-C. Limitaciones de nuestra aproximación

En algunos trabajos, como [24], se menciona la posibilidad de una arquitectura DTN para la IoNT. Con la aproximación DTN propuesta, sin embargo, en cada comunicación habría que incorporar la cabecera especificada por el protocolo Bundle a los datos y ello supone una complejidad inasumible En otros escenarios con restricciones en el volumen de datos a transmitir durante la comunicación, las cabeceras del protocolo Bundle supondrían también una limitación considerable.

IV. Seguridad DTN para la IoT

En está sección presentamos los mecanismos de seguridad de DTN y su adaptación a la IoT.

IV-A. Seguridad en DTN

El uso de la extensión de seguridad del protocolo Bundle provee de integridad y confidencialidad a los mensajes enviados de extremo a extremo. Por otra parte, la extensión de seguridad no especifica el tipo de claves criptográficas a usar. IBC ofrece una solución que se adapta a las necesidades de la arquitectura DTN.

IBC propone que la clave pública sea la propia identidad del usuario. Usando la identidad como clave pública, se pretende evitar tener que recurrir a una tercera parte de confianza que provea las claves necesarias. El esquema IBC, sin embargo, es incapaz de eliminarla. Debe existir al menos un generador de claves privadas (PKG, *Private Key Generator*). Resulta necesario que todos los nodos de la red se comuniquen con él para obtener las claves privadas correspondientes a su identidad.

Una aproximación jerárquica de IBC (HIBC, *Hirearchical Identity based cryptography*) [25] ofrece la escalabilidad necesaria tanto en arquitecturas DTN como en el paradigma de la IoT. Algunas implementaciones de HIBC como [26] proponen esquemas en los que el PKG genera las claves privadas para un conjunto de PKGs de nivel inferior, que serán los encargados de generar las claves privadas de un subconjunto de nodos de la red. Con esta propuesta, además de escalabilidad se garantiza que, en caso de quedar comprometido un PKG, no quede comprometido el sistema completo.

Cuando un nodo distinto del PKG queda comprometido es necesario revocar las claves. Sin embargo, dado que HIBC no necesita de una tercera parte para obtener las claves, no es posible comprobar que sigan siendo vigentes. En [27] se soluciona el problema de la revocación de claves mediante la concatenación de una marca de tiempo a la identidad en la generación de las claves públicas. Siendo así, la pareja de claves tiene una duración concreta y, en caso de que estas se vean comprometidas, únicamente lo estarán durante el periodo de tiempo en el que tienen validez.

Existen otras soluciones de seguridad que además de la confidencialidad, cubren problemas como el de la autenticación. En [28] se presenta una aproximación de clave simétrica donde, dado el esquema IBC propuesto por Boneh y Franklin [29], es posible establecer un secreto compartido entre cada pareja de nodos. Lo interesante de la propuesta reside en la posibilidad de generar el secreto compartido de forma no interactiva. De esta manera, en redes donde los contactos son oportunistas o donde existen limitaciones de cómputo, dos usuarios con las parejas de claves

$$(ID_U, d_U), (ID_V, d_V)$$

podrán calcular el secreto compartido entre ambos de forma independiente como:

$$K_{UV} = e(Q_U, d_V) = e(Q_V, d_U) = e(Q_U, Q_V)^s$$

donde s es el secreto maestro que únicamente conoce el PKG y $Q_U = H(ID_U)$ i $Q_V = H(ID_V)$ siendo H una función resumen. La igualdad expuesta, implica que el PKG tiene acceso al contenido de las comunicaciones, aún cuando dos nodos se hayan autenticado mutuamente y establecido un canal seguro.

IV-B. Adaptación a la IoT

Dado el requisito de escalabilidad que se debe cumplir en la IoT, proponemos como solución un sistema jerárquico de IBC como el propuesto en [26] que permite la creación de regiones, con un PKG por región y un nodo central que genera las claves para los PKG de cada una de las regiones. Junto con el sistema HIBC consideramos una marca temporal para gestionar la revocación de claves. Aunque el uso de marcas de tiempo da robustez y soluciona el problema de la revocación, supone que los nodos de la red deberán almacenar N claves. Sin embargo, en algunas ocasiones los nodos de la IoT tendrán

restricciones de memoria y ello podría generarles una excesiva dependencia con el PKG.

Tratando de solventar la dependencia que podría llegar a generarse con el PKG, y tratando de evitar también la capacidad de computo que se puede requerir para generar el canal seguro mediante el cifrado de cada mensaje con claves IBC, proponemos un esquema de clave simétrica basado en HIBC. La generación de claves simétricas para la autenticación mutua permite la creación de un canal seguro y auténtico sin la necesidad de usar las claves asimétricas. El uso exclusivamente privado de las claves IBC permitiría alargar su vida útil y reducir la dependencia del nodo con el PKG. Además, una vez calculada la clave, las operaciones para cifrar y descifrar no supondrán el mismo coste computacional que supondrían las operaciones en curvas elípticas en las que se basa IBC. Conseguimos entonces una reducción de la energía consumida por los nodos. Si tenemos en cuenta la propuesta [28], sería posible la generación de la clave simétrica de manera no interactiva, permitiendo un canal seguro y auténtico sin intercambio previo a la comunicación de datos.

Con la propuesta de la clave simétrica pretendemos también dotar de cierta protección contra la impersonalización en la solicitud de claves al PKG. Si consideramos que en la IoT los nodos que se agregan a una región no tienen que provenir de fuentes de confianza, estos podrían tomar la identidad de otro nodo frente al PKG y solicitar claves privadas. Con dicha solicitud conseguirían las claves privadas de otro nodo, pudiendo acceder a sus mensajes o actuar de forma deshonesta en su nombre.

Si asumimos que el PKG es habitualmente un nodo con más recursos de cómputo y memoria que el resto de los nodos se puede considerar la siguiente aproximación:

- Cuando un nodo (N) solicita claves por primera vez ante el PKG este comprueba si ya ha provisto de claves al nodo solicitante. Si es la primera solicitud, almacena la identidad del nodo y genera las claves correspondientes.
- Al solicitar claves de nuevo, el nodo envía al PKG la tupla (ID_N, $E_{K_{NPKG}}(ID_N)$).
- El PKG extrae el timestamp de ID_N y calcula el secreto compartido K_{NPKG}, entonces si $D_{K_{NPKG}}(ID_N) = ID_N$ genera la claves y las remite al nodo.

donde $E_{K_{NPKG}}(ID_N)$ es la identidad del nodo cifrada haciendo uso del secreto compartido entre el PKG y el nodo y K_{NPKG} es el secreto compartido. Es necesario recordar que el PKG no precisa de clave privada para realizar el cálculo del secreto compartido (véase IV-A). La extracción de la marca de tiempo de la identidad provista se plantea necesario debido a que en el momento de solicitar nuevas claves es posible que aquellas que almacena el nodo ya no tengan validez. La generación de la clave simétrica por parte del PKG, sin embargo, no requiere de la clave privada correspondiente a una determinada identidad. Gracias a esta característica es posible para el PKG validar la clave simétrica aún cuando la clave pública correspondiente haya caducado.

V. Escenario de aplicación

En esta sección se plantea un escenario de aplicación típico de la IoT y la aproximación DTN que le correspondería. Se pretende mostrar la adaptación de la aproximación propuesta para la IoT en un escenario real.

Un escenario típico en el campo de la IoT es la monitorización de la salud del usuario mediante pulseras o prendas de ropa con los sensores adecuados. Con la monitorización de la salud en mente, se plantea un escenario en el que diversos grupos de bomberos intentan apagar un incendio forestal. Los bomberos van equipados con trajes capaces de monitorizar sus constantes vitales y sus signos de fatiga. En el bosque no hay posibilidad de conseguir una conexión a Internet para mandar los datos al centro de mando.

Los vehículos que ayudan en las tareas de extinción cuentan con dispositivos que soportan la comunicación mediante el protocolo Bundle y que pueden entrar en comunicación con las unidades terrestres cuando estas se encuentran dentro del radio de acción del dispositivo. Los vehículos captan los datos de las unidades terrestres y al ir a recargar las cubas transmiten la información al centro de mando. Entonces los datos son procesadas y se pueden mandar ordenes de vuelta.

En este escenario los objetos, los trajes de los bomberos, son capaces de comunicarse con la central mediante el uso del protocolo Bundle y transmitir los datos adecuados para la correcta gestión de las unidades durante el incendio. Además, los trajes son capaces de recibir ordenes y transmitirlas a sus usuarios según lo considere el centro de coordinación.

El uso del protocolo Bundle junto con un sistema de seguridad resulta imprescindible. El protocolo Bundle asegura que se pueda producir la comunicación extremo a extremo, incluso cuando las conexiones con los vehículos son esporádicas. La seguridad garantiza que ninguna persona no autorizada pueda tener acceso a los datos de los bomberos o a las ordenes de la central. Para ello, los trajes de los bomberos pueden contar con claves simétricas creadas a partir de su número de identificación dentro del cuerpo de bomberos y de la identificación del vehículo, a partir de la matricula por ejemplo. El uso de las claves simétricas pre-computadas permitiría el intercambio rápido de mensajes en el tiempo limitado en que un camión o hidroavión se encontrase cerca de una unidad de extinción. Si los vehículos actúan, además, como nodos de confianza se pueden renovar las claves periódicamente gracias a una organización HIBC.

La aproximación DTN al escenario de la IoT permite la monitorización de los cuerpos de extinción, aún cuando estos se encuentren dispersos y fuera del alcance de comunicación con los vehículos. De esta manera es posible una gestión completa del operativo de forma segura.

VI. Conclusiones y trabajo futuro

En este trabajo hemos presentado una visión de la IoT y alguna de las medidas de seguridad que se han estudiado para el paradigma de la IoT. Hemos presentado, además, una aproximación DTN para la comunicación en la IoT y hemos revisado alguno de los mecanismos de seguridad que existen

para la arquitectura DTN, y que pueden ser adaptados a la IoT. Hemos considerado también alguna de las limitaciones existentes para la aproximación propuesta.

Como trabajo futuro deberíamos adaptar la propuesta de uso de las claves simétricas a la aproximación jerárquica de IBC para poder permitir el establecimiento de un canal seguro entre nodos pertenecientes a distintas regiones. Además, el cambio de región de un nodo no debería suponer un problema para obtener claves de un nuevo PKG. La posibilidad de realizar la solicitud de claves a otro PKG está estrechamente relacionada con la adaptación del mecanismo de identificación mutua en HIBC.

Sería también una línea de trabajo interesante estudiar la posibilidad de adaptar el protocolo Bundle para situaciones en que el sobrecoste de enviar las cabeceras hace inviable el uso de una aproximación DTN, por ejemplo en el caso de nano-dispositivos.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el Ministerio de Ciencia e Innovación Español, proyecto TIN2010-15764 y por la Generalitat de Cataluña, proyecto 2014SGR-619.

REFERENCIAS

[1] R. H. Weber and R. Weber, *Internet of Things*. Springer, 2010.
[2] I. F. Akyildiz and J. M. Jornet, "The internet of nano-things," *Wireless Communications, IEEE*, vol. 17, no. 6, pp. 58–63, 2010.
[3] M. A. Feki, F. Kawsar, M. Boussard, and L. Trappeniers, "The internet of things: The next technological revolution," *Computer*, vol. 46, no. 2, pp. 24–25, 2013.
[4] K. Fall, "A delay-tolerant network architecture for challenged internets," in *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pp. 27–34, ACM, 2003.
[5] K. Finkenzeller, *RFID Handbook*. Wiley Online Library, 2003.
[6] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393–422, 2002.
[7] S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, *Mobile ad hoc networking*. John Wiley & Sons, 2004.
[8] G. Hancke, K. Markantonakis, and K. Mayes, "Security challenges for user-oriented RFID applications within the Internet of Things," *Journal of Internet Technology*, vol. 11, no. 3, pp. 307–313, 2010.
[9] A. Juels, "RFID security and privacy: A research survey," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 381–394, 2006.
[10] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and privacy in distributed Internet of Things," *Computer Networks*, vol. 57, no. 10, pp. 2266–2279, 2013.
[11] ZigBee Alliance, "Zigbee specification," *URL: http://www. zigbee. org*, vol. 558, 2006.
[12] G. Mulligan, "The 6LoWPAN architecture," in *Proceedings of the 4th workshop on Embedded networked sensors*, pp. 78–82, ACM, 2007.
[13] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer networks*, vol. 47, no. 4, pp. 445–487, 2005.
[14] C. Alcaraz and J. Lopez, "A security analysis for wireless sensor mesh networks in highly critical systems," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 40, no. 4, pp. 419–428, 2010.
[15] P. Szczechowiak, A. Kargl, M. Scott, and M. Collier, "On the application of pairing based cryptography to wireless sensor networks," in *Proceedings of the second ACM conference on Wireless network security*, pp. 1–12, ACM, 2009.
[16] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in cryptology*, pp. 47–53, Springer, 1985.
[17] G. Zhao, X. Si, J. Wang, X. Long, and T. Hu, "A novel mutual authentication scheme for internet of things," in *Modelling, Identification and Control (ICMIC), Proceedings of 2011 International Conference on*, pp. 563–566, IEEE, 2011.
[18] A. Alcaide, E. Palomar, J. Montero-Castillo, and A. Ribagorda, "Anonymous authentication for privacy-preserving IoT target-driven applications," *Computers & Security*, vol. 37, pp. 111–123, 2013.
[19] X. J. Lin and L. Sun, "Insecurity of an anonymous authentication for privacy-preserving IoT target-driven applications," *IACR Cryptology ePrint Archive*, vol. 2013, p. 795, 2013.
[20] K. L. Scott and S. Burleigh, "Bundle protocol specification," 2007.
[21] C. Borrego and S. Robles, "A store-carry-process-and-forward paradigm for intelligent sensor grids," *Information Sciences*, vol. 222, no. 0, pp. 113 – 125, 2013.
[22] C. Borrego, S. Castillo, and S. Robles, "Striving for sensing: Taming your mobile code to share a robot sensor network," *Information Sciences*, 2014.
[23] S. Symington, S. Farrell, H. Weiss, and P. Lovell, "Bundle security protocol specification," *Work Progress, October*, 2007.
[24] S. Balasubramaniam and J. Kangasharju, "Realizing the internet of nano things: challenges, solutions, and applications," *Computer*, vol. 46, no. 2, pp. 62–68, 2013.
[25] C. Gentry and A. Silverberg, "Hierarchical ID-based cryptography," in *Advances in cryptology-ASIACRYPT 2002*, pp. 548–566, Springer, 2002.
[26] R. Patra, S. Surana, and S. Nedevschi, "Hierarchical identity based cryptography for end-to-end security in DTNs," in *Intelligent Computer Communication and Processing, 2008. ICCP 2008. 4th International Conference on*, pp. 223–230, IEEE, 2008.
[27] A. Seth and S. Keshav, "Practical security for disconnected nodes," in *Secure Network Protocols, 2005.(NPSec). 1st IEEE ICNP Workshop on*, pp. 31–36, IEEE, 2005.
[28] R. Dupont and A. Enge, "Provably secure non-interactive key distribution based on pairings," *Discrete Applied Mathematics*, vol. 154, no. 2, pp. 270–276, 2006.
[29] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology-CRYPTO 2001*, pp. 213–229, Springer, 2001.

Smart-Shopping: Aplicación de un Protocolo de Firma de Contratos Multi-Two-Party Atómico

Gerard Draper-Gil, Josep-Lluís Ferrer-Gomila, M. Francisca Hinarejos
Universitat de les Illes Balears (UIB), Email: {gerard.draper, jlferrer, xisca.hinarejos}@uib.es

***Resumen*—El avance de Internet y las tecnologías de comunicaciones está disminuyendo cada vez más la distancia entre consumidores y proveedores, hasta el punto que cualquier proveedor que lo desee puede ofrecer sus productos directamente al consumidor final. Esto supone a la vez una ventaja y una desventaja para el consumidor. Por un lado, le permite comparar los precios de distintos proveedores, pero por otra parte la gran cantidad de oferta puede complicar este proceso. Un caso particularmente interesante es la situación en la que el consumidor quiera un producto multi servicio, como los paquetes turísticos, formados por vuelos, hoteles, excursiones, etc.**

En este artículo presentamos una modificación sobre un protocolo multi-two-party atómico, que permite al consumidor automatizar la función búsqueda, negociación y compra (firma de un contrato), manteniendo la equitatividad y atomicidad en la transacción.

***Palabras clave*—Contratación electrónica multi- party, E- Commerce, Smart Shopping**

I. Introducción

Gracias al comercio electrónico, hoy en día consumidores y proveedores están más cerca que nunca. A través de Internet, los consumidores tienen acceso directo a múltiples proveedores, permitiéndoles, entre otras cosas, comparar distintas ofertas y quedarse con la que más les interese. A su vez, los proveedores tienen acceso directo a millones de clientes potenciales.

Esta situación es de especial interés en sectores como el ocio, donde los consumidores adquieren habitualmente productos como los paquetes turísticos, formados por varios servicios: hoteles, vuelos, excursiones, etc. Los consumidores pueden fácilmente comparar el precio ofrecido para un mismo servicio por distintos proveedores (existen incluso webs específicas para estos servicios) y escoger el que más les convenga. Al final, el paquete que compra el consumidor puede estar formado por servicios de diferentes proveedores. El problema aparece en el momento de ejecutar la compra de los servicios: para que el consumidor obtenga el producto que desea, tiene que comprar servicios diferentes de proveedores distintos, por lo tanto, debe comprometerse con todos los proveedores o con ninguno; sino fuera así, su paquete no estaría completo. A este tipo de escenarios se les denomina Multi-two-Party Atómicos (AM2P).

En un escenario Multi-Two-Party (M2P) tenemos N participantes, 1 consumidor C y $(N-1)$ proveedores P_i, agrupados en un conjunto de $(N-1)$ pares $\{C, P_1\}, \{C, P_2\}..., \{C, P_{(N-1)}\}$, que quieren firmar un conjunto de $(N-1)$ contratos $\{M_1, M_2, ..., M_{(N-1)}\}$ dos a dos, es decir, C y P_1 quieren firmar el contrato M_1, C y P_2 el contrato M_2, etc. En este escenario, ni C ni P_i quieren dar su firma sin tener la seguridad que el otro participante enviará la suya. El escenario *Multi-Two-Party Atómico* (AM2P) es un caso restrictivo del Multi-Two-Party en el que C no quiere enviar su firma sin tener la seguridad que recibirá la firma de *todos* los proveedores $\{P_1, ..., P_{(N-1)}\}$, ni P_i quiere enviar la suya si no recibe la correspondiente firma de C sobre el contrato M_i.

Un artículo presentado en la anterior edición de la RECSI [1] presenta el primer protocolo de firma de contratos dirigido a estos escenarios, donde el consumidor debe negociar previamente con todos los proveedores antes de ejecutar el protocolo. Es decir, el objetivo del protocolo es firmar una serie de contratos pre-acordados entre el consumidor y los distintos proveedores. Este proceso puede ser largo y tedioso, y no todos los consumidores tienen el tiempo o los conocimientos para llevarlo a cabo. Para solucionar esta situación, en este artículo presentamos una modificación sobre el protocolo AM2P, que permite fusionar las fases de negociación y firma, facilitando su uso a los consumidores.

Contribución: En este artículo presentamos una propuesta de protocolo para firma digital de contratos para escenarios *Multi-Two-Party Atómicos*, donde la fase de negociación forma parte del proceso de firma. Esta propuesta es una modificación sobre un protocolo presentado en la última RESCI [1], manteniendo los requisitos de seguridad: efectividad, equitatividad, temporalidad, no-repudio, confidencialidad y verificabilidad de la TTP.

Organización: El artículo está organizado de la siguiente manera. La sección II presenta un ejemplo de cómo podría utilizarse el protocolo presentado en este artículo para crear una aplicación de compra inteligente. En la sección III se describen los requisitos de seguridad del protocolo de negociación más firma *Multi-Two-Party Atómico*. En la sección IV se discute brevemente el trabajo previo realizado, y se presenta el protocolo de firma *Multi-Two-Party Atómico* en el cual se basa este artículo. Nuestra propuesta se define en la sección V. En la sección VI analizamos si nuestra propuesta cumple con los requisitos de seguridad. Finalmente, las conclusiones aparecen en la sección VII.

II. Escenario

El protocolo propuesto en [1], permite a los consumidores firmar un conjunto de contratos, cada uno con un proveedor distinto, de manera atómica y equitativa. Si lo llevamos al terreno práctico, para poder ejecutar el protocolo, el consumidor

requiere de una aplicación, o bien nativa o bien como servicio. A esta aplicación el consumidor debería facilitarle un conjunto de contratos, que previamente tiene que haber negociado. Esta fase de negociación previa puede suponer un problema. No todos los consumidores tendrán el tiempo o conocimientos necesarios para llevarla a cabo. La modificación que proponemos sobre el protocolo presentado en [1], permitiría a la aplicación del consumidor automatizar las tareas de búsqueda de proveedores, negociación y firma de contratos.

En la figura 1 se muestra un ejemplo de cómo podría utilizarse el protocolo para implementar una aplicación de compras inteligente. El funcionamiento sería el siguiente:

1. El consumidor le indica a la aplicación que quiere comprar un paquete compuesto por un servicio aéreo y un alojamiento (orden de compra). Para cada uno de los servicios le indica las opciones que desea. Por ejemplo, en el caso del servicio aéreo, las fechas del viaje, origen y destino, horarios, etc... Además, el consumidor puede indicar un precio máximo para el paquete completo, sus preferencias en caso de que el producto sea ofrecido por más de un proveedor, incluso podría indicar un tiempo màximo en el que la aplicación debe contestar (inmediatamente, un día, una semana, ...).
2. La aplicación consultará su base de datos de proveedores y recuperará la lista de proveedores que puedan ofrecer los servicios reclamados por el consumidor. Esta base de datos puede ser un servicio preconfigurado en la aplicación, un listado que el propio consumidor haya confeccionado, un servicio externo (por ejemplo UDDI [2]), etc.
3. Una vez se ha generado la lista de proveedores, se inicia el protocolo de negociación + firma con cada uno de ellos. En el caso del ejemplo, se han encontrado 3 proveedores de servicios aéreos y 2 proveedores de servicios de alojamiento.
4. Como resultado, la aplicación de compras le devuelve al consumidor el contrato firmado con cada uno de los proveedores seleccionados (para firmar), en este caso han sido el proveedor P_{V1} de servicios aéreos y el proveedor P_{H1} de servicios de alojamiento.

¿Cómo se introduce la negociación en el proceso de firma?, veamos el caso del ejemplo de la figura 1. La aplicación de compra preparará un contrato para cada uno de los proveedores que ha encontrado en la base de datos: $\{M_{P_{V1}}, M_{P_{V2}}, M_{P_{V3}}, M_{P_{H1}}, M_{P_{H2}}\}$, y lanzará la petición a todos ellos. Supongamos que los proveedores P_{V1} y P_{V2} aceptan la petición y el proveedor P_{V3} la rechaza. En este caso la aplicación de compras deberá escoger entre P_{V1} y P_{V2} para continuar con la ejecución, mientras que al proveedor descartado deberá enviarle un mensaje de rechazo. Los criterios a seguir pueden ser varios, por ejemplo las preferencias del consumidor (puede indicar proveedores favoritos), el tiempo de respuesta, o un valor de reputación. Pero antes de poder contestar al proveedor de servicios aéreos, el consumidor debe recibir al menos una respuesta válida de un proveedor de

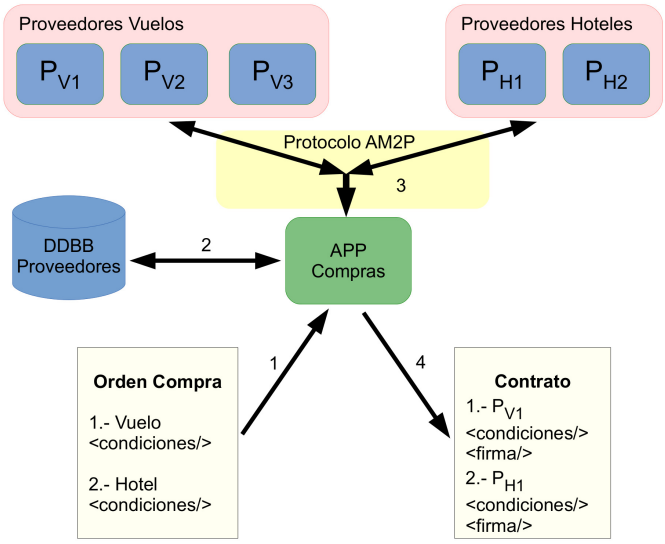


Figura 1. Ejemplo de Aplicación

alojamiento. Una vez el consumidor tiene confirmación de que todos los servicios incluidos en la orden de compra están disponibles, puede continuar con la ejecución del protocolo.

III. Requisitos de Seguridad

Asokan *et al.* [3] y Zhou *et al.* [4] establecen los requisitos mínimos para el intercambio equitativo: efectividad, equitatividad, temporalidad, no-repudio y verificabilidad de la TTP. Aunque de hecho, la verificabilidad de la TTP no es estríctamente necesaria para que un protocolo sea equitativo. Es más, Asokan *et al.* [3] define 2 tipos de equitatividad, débil y fuerte, mientras que Zhou *et al.* [4] define sólo uno, que coincide con la equitatividad fuerte. Otro requisito deseable es la confidencialidad. A continuación detallamos los requisitos para el protocolo equitativo Multi-Two-Party Atómico definido en este artículo:

- **Efectividad.** Si todas las partes involucradas en un protocolo de negociación más firma Multi-Two-Party Atómico se comportan correctamente, el consumidor recibirá la firma de los $(N-1)$ proveedores seleccionados, y estos recibirán la correspondiente firma del consumidor. Además, todos los proveedores que hayan sido descartados recibirán un mensaje de rechazo por parte del consumidor, y éste recibirá el correspondiente reconocimiento por parte del proveedor. En caso de que sea el proveedor quien rechaze la negociación, se lo indicará al consumidor. Todos estos mensajes se intercambiarán sin que intervenga la TTP.
- **Equitatividad Débil Multi-Two-Party Atómica.** Al finalizar una negociación más firma Multi-Two-Party Atómica, el consumidor honesto tendrá la firma de los $(N-1)$ proveedores escogidos, y los proveedores honestos tendrán la correspondiente firma o mensaje de rechazo del consumidor; o todas las partes honestas conseguirán

evidencias suficientes para demostrar, ante un árbitro, que se han comportado correctamente.

- **Temporalidad.** Todos los participantes en una negociación más firma Multi-Two-Party Atómica tienen la seguridad de que la ejecución del protocolo de firma tendrá una duración finita. Una vez finalizada, no se puede degradar el nivel de equitatividad obtenida por los participantes honestos, independientemente del comportamiento del resto de participantes.
- **No-Repudio.** En una negociación más firma Multi-Two-Party Atómica en la que hay involucrados un consumidor y $N-1$ proveedores seleccionados por el consumidor, ni el consumidor ni los proveedores pueden negar haber estado involucrados. En particular, dado un contrato firmado M_i, ni el consumidor C ni el proveedor P_i pueden negar haberlo firmado. Además, ninguno de los proveedores descartados puede negar haber sido rechazado por el consumidor, ni el consumidor en caso de que sea el proveedor quien rechaze una oferta.
- **Confidencialidad.** Sólo los participantes involucrados en una firma, es decir, el consumidor C y el proveedor P_i, pueden conocer el contenido del contrato M_i. Ni siquiera la TTP debe tener acceso al contrato en claro.
- **Verificabilidad de la TTP.** Si la TTP actúa de manera deshonesta, provocando la pérdida de equitatividad de un participante honesto (consumidor o proveedor), este puede probar el comportamiento deshonesto de la TTP frente a un árbitro externo.

IV. Trabajo Previo

Pese a los muchos esfuerzos dedicados al estudio del intercambio equitativo, existen muy pocas propuestas [5], [6] que traten el problema que presentamos en este artículo. De hecho, solo una de ellas [5] prevee una fase de negociación, aunque no está integrada en el protocolo de firma, ya que se trata de una fase previa.

En el protocolo de firma de contratos Multi-Two-Party Atómico presentado en [1], sobre el cual se basa la propuesta presentada en este artículo, se incluye una revisión de las propuestas similares existentes. Como conclusión, a dicha revisión y hasta donde conocemos, ninguna de las referencias que hemos encontrado en la literatura en relación con intercambios Atómicos Multi-Two-Party ([5], [6]) cumple con los requisitos de seguridad necesarios, para nuestro escenario (ver sección III).

A continuación se presenta la notación que se va a utilizar a lo largo del artículo:

- $\mathbf{N}$ Número de participantes en la fase de firma: 1 Consumidor y $N-1$ Proveedores.
- $\mathbf{NP}$ Número de proveedores participantes en la fase de negociación: $NP \geq N-1$.
- $\overline{\mathbf{X_Z}} = \{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_{(Z)}}\}$ Vector con Z elementos.
- $\overline{\mathbf{RP}}$ *Rejected Providers*, conjunto de proveedores que el consumidor rechazará.
- $\overline{\mathbf{AP}}$ *Accepted Providers*, conjunto de proveedores que el consumidor escogerá para continuar la ejecución del protocolo.
- $\mathbf{C}$ Consumidor.
- $\mathbf{P_i}$ Proveedor i, $1 \leq i \leq NP$.
- $\mathbf{M_i}$ Mensaje (contrato) intercambiado entre el consumidor C y el proveedor P_i.
- **CID** Identificador de Contrato Único (Unique Contract IDentifier).
- $\mathbf{h(M_i)}$ Función de Hash del mensaje M_i.
- $\mathbf{S_j[M_i]} = \mathbf{SK_j[h(M_i)]}$ Firma Digital de j sobre M_i (donde SK_j es la clave privada de j).

El protocolo presentado en este artículo está basado en una propuesta previa para la firma de contratos AM2P [1] con N participantes, 1 consumidor y $N-1$ proveedores. Se trata de un protocolo optimista con arquitectura en paralelo, donde el consumidor contacta con todos los proveedores "a la vez", y espera su respuesta antes de continuar con a ejecución, es decir, el consumidor envía $N-1$ compromisos (COMmitment) al mismo tiempo, y espera a recibir las $N-1$ aceptaciones (ACCeptance) antes de continuar. Si el consumidor deja de recibir una o más aceptaciones, contactará con la TTP. Los compromisos son los mensajes enviados desde el consumidor a los proveedores, y las aceptaciones son los mensajes enviados de los proveedores al consumidor. Los $COM_{(n,i)}$ y $ACC_{(n,i)}$ (n = número de ronda, i = número de proveedor) son las evidencias que el proveedor P_i y el consumidor C deben recibir, respectivamente.

La propuesta de protocolo optimista de firma electrónica de contratos AM2P [1] está dividida en dos sub-protocolos: *intercambio* y *resolución*. Si todas las partes involucradas se comportan correctamente, el sub-protocolo de *intercambio* terminará después de N rondas, se intercambiarán $2N(N-1)$ mensajes y la TTP no intervendrá.

Cada ejecución completa del sub-protocolo de *resolución* está compuesta de N rondas, y cada ronda requiere el intercambio de $N-1$ pares de mensajes {compromiso, aceptación}, lo que hace un total de $2N(N-1)$ mensajes. Las evidencias de firma son las correspondientes a la ronda N ($COM_{(N,i)}$ y $ACC_{(N,i)}$). En cualquier momento, el consumidor y los proveedores pueden ejecutar el sub-protocolo de *resolución* para resolver la ejecución del protocolo. Durante la primera ronda ($n=1$), cualquier participante puede contactar con la TTP y solicitar que se cancele la firma, mientras que si $n>1$, la petición tendrá como objetivo finalizar el protocolo (firmar el contrato).

V. Protocolo

El protocolo presentado en este artículo mantiene la estructura original, una arquitectura en paralelo donde el consumidor y los proveedores intercambian N pares de mensajes {compromiso, aceptación}.

V-A. Sub-Protocolo de Intercambio

En la tabla I podemos ver el flujo de ejecución del protocolo y los mensajes intercambiados. La fase de negociación se realiza durante la primera ronda y la mitad de la segunda. En la primera ronda, en lugar de contactar con $(N-1)$ proveedores,

Sub-Protocolo de Intercambio

Ronda			
1	$C \rightarrow P_i$	$CID, M_i, 1, COM_{(1,i)}$	$i \in [1..NP]$
1	$C \leftarrow P_i$	$CID, 1, ACC_{(1,i)}$	ACEPTACIÓN
		$CID, M_i, M_i^{'}, 1, ACC_{(1,i)}^{'}$	CONTRAOFERTA
		$CID, 1, REJ_{(1,i)}$	RECHAZO
2a	$C \rightarrow P_r$	$CID, 2, REJ_{(2,r)}$	$r \in \overline{RP}$
	$C \leftarrow P_r$	$CID, 2, ACKR_{(2,r)}$	
2b	$C \rightarrow P_i$	$CID, 2, COM_{(2,i)}$	$i \in \overline{AP}$
	$C \leftarrow P_i$	$CID, 2, ACC_{(2,i)}$	
⋮	⋮	⋮	
n	$C \rightarrow P_i$	$CID, n, COM_{(n,i)}$	
n	$C \leftarrow P_i$	$CID, n, ACC_{(n,i)}$	
⋮	⋮	⋮	
N	$C \rightarrow P_i$	$CID, N, COM_{(N,i)}$	
N	$C \leftarrow P_i$	$CID, N, ACC_{(N,i)}$	

$COM_{(n,i)} = S_C[CID, h(M_i), n]$
$ACC_{(n,i)} = S_{P_i}[CID, h(M_i), n]$
$ACC^{'}_{(n,i)} = S_{P_i}[CID, h(M_i), h(M_i^{'}), n]$

$ACKR_{(2,r)} = S_{P_r}[CID, h(M_r), 2, ACK - REJECTED]$
$REJ_{(z,i)} = S_{X_i}[CID, h(M_i), z, REJECTED]$;
$z = 1 \rightarrow X_i = P_i$; $z = 2 \rightarrow X_i = C$

Tabla I
SUB-PROTOCOLO DE INTERCAMBIO ATÓMICO MULTI-TWO-PARTY

Sub-Protocolo de Resolución

Consumidor $\mathbf{peticionResolucion_{(n,i)}}$
$CID, h(M_i), n$
$COM_{(1,1)}, ACC_{(1,1)}, ..., COM_{(1,(N-1))}, ACC_{(1,(N-1))}$
⋮ ⋮ ⋮
$COM_{(n,1)}, ACC_{(n,1)}, ..., COM_{(n,i)}, EVRES_{(n,i)}$

Proveedor $\mathbf{P_i}$ $\mathbf{peticionResolucion_{(n,i)}}$
$CID, h(M_i), n$
$COM_{(1,i)}, ACC_{(1,i)}, ..., COM_{(n,i)}, ACC_{(n,i)}, EVRES_{(n,i)}$

TTP $\mathbf{RespuestaResolucionCancelada_{(n,i)}}$
$Canceled_{TK} = S_{TTP}[CID, h(M_i), n, canceled]$

TTP $\mathbf{RespuestaResolucionFirmada_{(n,i)}}$
Consumer $Signed_{TK} = S_{TTP}[CID, h(M_i), n, COM_{(n,i)}]$
Provider $Signed_{TK} = S_{TTP}[CID, h(M_i), n, ACC_{(n,i)}]$

N = número de participantes; n = ronda
$COM_{(n,i)} = S_C[CID, h(M_i), n]$
$ACC_{(n,i)} = S_{P_i}[CID, h(M_i), n]$
$EVRES_{(n,i)} = S_{(C\ or\ P_i)}[CID, h(M_i), n, ...]$, firma sobre el mensaje enviado.

Tabla II
SUB-PROTOCOLO DE RESOLUCIÓN MULTI-TWO-PARTY ATÓMICO

el consumidor contactará con un número $NP \geq (N-1)$, y el mensaje M_i enviado será una propuesta de contrato. Los proveedores podrán rechazar la oferta (REJ), aceptar la oferta (ACC), o enviar una contraoferta ($ACC^{'}$). Una vez el consumidor ha recibido suficientes respuestas (al menos una positiva por servicio que desee), escogerá entre ellas las $(N-1)$ que más le interesen para continuar la ejecución del protocolo. La siguiente ronda (la número 2) se ejecutará en dos fases. Primero ($2a$), el consumidor informará a los proveedores P_r ($r \in \overline{RP}$) que no hayan sido escogidos, enviándoles un mensaje REJ, y éstos contestarán indicando que han recibido el mensaje, $ACKR$. A continuación ($2b$), el consumidor esperará a recibir todas las respuestas de los P_r descartados. Por cada P_r que no conteste, el consumidor enviará una petición a la TTP para informar de que estos proveedores han sido descartados (ver tabla III). En la segunda parte de la segunda ronda (ver ronda 2b en tabla I), el consumidor continuará la ejecución del protocolo con los $(N-1)$ proveedores escogidos (P_i $i \in \overline{AP}$), intercambiando mensajes de compromiso y aceptación, hasta conseguir el compromiso correspondiente a la ronda N, evidencia de firma.

V-B. Sub-Protocolo de Resolución

Al igual que en el protocolo AM2P original [1], consumidor y proveedores pueden contactar con la TTP en cualquier momento. Durante la primera ronda ($n = 1$), cualquier participante puede contactar con la TTP y solicitar que se cancele la firma, mientras que si $n > 1$, la petición tendrá como objetivo finalizar el protocolo (firmar el contrato). El subprotocolo de resolución (tabla II) es igual al del protocolo original [1], con una función añadida, el informe por parte del consumidor de que un proveedor P_r ha sido descartado. En la tabla III vemos el mensaje de petición y respuesta del informe de reachazo. Esta petición es necesaria para evitar que un P_r ($r \in \overline{RP}$) que ha sido rechazado por el cliente pueda obtener una evidencia de firma de la TTP y forzar su cumplimiento.

Al igual que en el protocolo orignial [1], la TTP utiliza un conjunto de reglas para solucionar correctamente las peticiones de *resolución* recibidas. Estas reglas deben aplicarse en un cierto orden, como se muestra a continuación:

- **R0** La TTP sólo aceptará una petición de resolución por participante y CID. En el caso del consumidor, las peticiones de rechazo no se contabilizarán.
- **R1** Si la TTP recibe una petición de un participante X_i durante la ronda $n = 1$, y la ejecución no ha sido previamente finalizada (*signed=true*) por otro participante ni X_i ha sido informado como rechazado por el consumidor C, la TTP cancelará la firma y le enviará a X_i una prueba de que la firma ha sido cancelada.
- **R2** Si la TTP recibe una petición de X_i durante la ronda $n > 1$, y la ejecución no ha sido previamente cancelada por otro participante ni X_i ha sido informado como rechazado por el consumidor C, la TTP la finalizará (*signed=true*) y le enviará a X_i una prueba de que el contrato está firmado.
- **R3** Si la TTP recibe una petición de X_i durante la ronda $n = 1$, y la ejecución ha sido previamente finalizada (*signed=true*) por otro participante y X_i no ha sido informado como rechazado por el consumidor C, la

Sub-Protocolo de Resolución: Informe Rechazo	
$C \rightarrow TTP$	$CID, h(M_i), COM_{(1,1)}, REJ_{(2,r)}$
$C \leftarrow TTP$	$ACK_{TK} = S_{TTP}[CID, h(M_i), 2, REJECTED]$

Tabla III
SUB-PROTOCOLO DE RESOLUCIÓN MULTI-TWO-PARTY ATÓMICO, INFORME DE RECHAZO

TTP enviará a X_i una prueba de que el contrato está firmado.

R4 Si la TTP recibe una petición de X_i durante una ronda $n > 1$, y la ejecución ha sido previamente cancelada por otro participante, la TTP revisará las peticiones previamente recibidas para comprobar si alguien ha hecho trampas. Si la TTP decide que todas las peticiones anteriores eran incorrectas, cambiará el estado de la ejecución a *signed=true* y enviará la correspondiente prueba de firma a X_i. De lo contrario, el estado continuará siendo *canceled=true* y la TTP enviará a X_i la correspondiente prueba de cancelación.

VI. REVISIÓN DE SEGURIDAD

En esta sección comprobaremos si nuestra propuesta cumple con los requisitos de seguridad para protocolos de Intercambio Equitativo Multi-Two-Party Atómicos, aplicados a la firma digital de contratos, definidos en la sección III: efectividad, equitatividad, temporalidad, no-repudio, verificabilidad de la TTP y confidencialidad.

Efectividad. La ejecución del sub-protocolo de *intercambio* (tabla I) nos asegura que, si todos los participantes actúan correctamente, el consumidor recibirá la firma de los $N-1$ proveedores escogidos, y cada proveedor $P_i, i \in \overline{AP}$ recibirá su correspondiente firma del consumidor C después de N rondas y sin intervención de la TTP. Además, todos los proveedores descartados $P_r, r \in \overline{RP}$ recibirán evidencia de que no participa en la firma del contrato M_r. Por lo tanto, el protocolo cumple con el requisito de efectividad.

Equitatividad Débil Multi-Two-Party Atómica. Si consideramos al consumidor honesto, con independencia del comportamiento del proveedor, el consumidor mantendrá la equitatividad. Hay tres posibilidades en las que un proveedor P_i (con $1 \leq i \leq (N-1)$) puede obtener una prueba de firma del consumidor:

- Después de recibir el N-ésimo compromiso $COM_{(N,i)}$, $(1 < i < (N-1))$, lo que significa que el consumidor tiene $N-1$ aceptaciones del proveedor, con lo que puede contactar con la TTP y obtener una evidencia de firma.
- Después de contactar con la TTP, lo que implica que la variable *signed* es igual a *true*, por lo tanto, el consumidor puede obtener una evidencia de firma del proveedor o de la TTP (aplicando **R3**), si el proveedor decide no continuar la secuencia de N rondas.
- Un proveedor descartado podría hacer trampas y conseguir una firma de la TTP, siguiento el ejemplo de abort-chaining (explicado en [1]). Pero si ha sido descartado por el consumidor, este tendrá o bien el mensaje $ACKR$ del propio consumidor, o el ACK_{TK} de la TTP, con lo que podrá demostrar que el proveedor hizo trampas.

En ambas situaciones, el consumidor mantiene la equitatividad.

Si consideramos un proveedor honesto P_i $(1 \leq i \leq (N-1))$, con independencia del comportamiento del consumidor, el proveedor mantendrá la equitatividad. El consumidor puede obtener una prueba de firma del proveedor de tres maneras distintas:

- Después de recibir la N-ésima aceptación del proveedor $ACC_{(N,i)}$, lo que implica que el proveedor ya tiene la prueba de firma del consumidor $COM_{(N,i)}$.
- Contactando con la TTP en la ronda $n > 1$ (aplicando **R2**), lo que quiere decir que la TTP tiene la variable *signed = true*. Por lo tanto, el proveedor podrá obtener la prueba de firma del mismo consumidor, o de la TTP (aplicando **R3**) si el consumidor decide interrumpir la secuencia de N rondas.
- Un consumidor tramposo puede descartar a un proveedor (REJ) y luego contactar con la TTP reclamando la firma del contrato con ese proveedor (aplicando **R2**). En este caso, la firma obtenida por el consumidor no tendría validez, puesto que el proveedor puede demostrar que fue rechazado, utilizando el mensaje REJ recibido.

En todas las situaciones el proveedor mantiene la equitatividad. Por lo tanto, podemos afirmar que el protocolo cumple con el requisito de Equitatividad Débil Multi-Two-Party Atómica.

Temporalidad. En cualquier momento durante la ejecución del protocolo, cualquier participante puede ejecutar el sub-protocolo de *resolución* y finalizar su ejecución, obteniendo una prueba o bien de firma, o bien de cancelación. Si todos los participantes se comportan de manera correcta el protocolo requiere de N rondas y $4NP + 2(N-1)(N-2)$ mensajes, siendo N un número finito y conocido. Por lo tanto, podemos afirmar que el protocolo tiene una duración finita, ya sea porque interviene la TTP, o por la ejecución normal de este. Es más, una vez el protocolo ha terminado, su estado final no puede cambiar. Si el protocolo finaliza con la intervención de la TTP, esta se encargará de mantener la coherencia entre las distintas peticiones posibles recibidas (siguiendo las reglas de la TTP). Si el protocolo ha finalizado después de la N-ésima ronda, las evidencias obtenidas por proveedor y consumidor servirán como prueba de su estado final. Por tanto, podemos afirmar que el protocolo cumple con el requisito de temporalidad.

No-repudio. Durante la negociación+firma de un contrato Multi-Two-Party Atómico, se generan, en cada ronda, evidencias de la participación del consumidor y de los proveedores. Por un lado tenemos los mensajes $COM_{(n,i)}$ y

los $ACC_{(n,i)}$, que relacionan a consumidores y proveedores con la firma de un contrato, y por otra parte los mensajes REJ y $ACKR$ que prueban lo contrario. En particular, el N-ésimo compromiso y la N-ésima aceptación, son considerados como la firma del contrato. Si un consumidor intenta desvincularse de la firma de un contrato M_i con el proveedor P_i, este puede probar la implicación del consumidor utilizando la firma realizada por el propio consumidor, o una evidencia obtenida de la TTP. De la misma manera, si el proveedor intenta desvincularse, el consumidor puede probar su implicación utilizando la firma generada por el proveedor, o las evidencias recibidas de la TTP.

Verificabilidad. La TTP puede comportarse de forma deshonesta y generar evidencias erróneas, dando como resultado, que algún participante honesto pueda perder su equitatividad. Suponiendo que el consumidor es honesto y la TTP deshonesta, pueden darse las siguientes situaciones:

- El consumidor envía una petición de *resolución* en la ronda $n = 1$, y la TTP contesta con una evidencia de firma. De acuerdo a las reglas de la TTP, durante la ronda $n = 1$ los participantes sólo pueden obtener prueba de cancelación. Por lo que el consumidor sabrá que la TTP ha enviado una respuesta equivocada (el consumidor no ha enviado ningún mensaje de ronda 2). Para probarlo, el consumidor puede pedirle a la TTP que presente las pruebas de la petición de resolución recibida previamente, esto es, pruebas de la ronda 2.
- El consumidor envía una petición durante la ronda $n \geq 2$ porque uno o más proveedores no han enviado su mensaje de aceptación, y la TTP responde con una evidencia de cancelación (sin que haya habido una cancelación previa), o de firma (habiendo recibido una cancelación previa). Como estamos en la ronda $n \geq 2$, las reglas de la TTP establecen que la respuesta debe ser una evidencia de firma a no ser que algún otro participante la haya cancelado, por lo tanto, cualquier respuesta es válida. Pero las evidencias contradictorias que la TTP haya enviado al consumidor y a uno o más proveedores probará la irregularidad en el comportamiento de la TTP.
- La TTP envía una evidencia de firma a un proveedor que ha sido previamente descartado (el consumidor ha informado a la TTP) por el consumidor. En este caso el consumidor podrá demostrar que la TTP ha emitido evidencias erróneas, presentando el ACK_{TK} recibido como respuesta al mensaje de informe de rechazo.

Suponiendo un proveedor honesto y la TTP deshonesta, puede darse la siguiente situación:

- El proveedor envía una petición de *resolución* durante la ejecución de la ronda n porque no ha recibido el compromiso de la ronda $(n+1)$ y la TTP responde con una cancelación (sin que haya habido una cancelación previa), o una firma (habiendo recibido una cancelación previa). Ambos resultados son coherentes con las reglas de la TTP, pero en ambos casos el proveedor y el consumidor tendrán evidencias contradictorias enviadas y firmadas por la TTP, lo que probará su mal comportamiento.
- La TTP envía una evidencia de firma a un consumidor, al que el proveedor ha rechazado previamente. Si el consumidor intenta forzar la ejecución del contrato, el proveedor podrá reclamar a la TTP que presente evidencias de aceptación, el mensaje $ACC_{(1,i)}$. Como el proveedor no ha enviado nunca este mensaje (ha enviado un $REJ_{(1,i)}$), podrá demostrar que la TTP y el consumidor se han comportado deshonestamente.

Confidencialidad. La ejecución del sub-protocolo de *resolución* no requiere el envío del texto en claro del contrato (M_i), es decir, sin cifrar. La TTP sólo recibe el resultado de aplicar una función de "hash" sobre M_i. Además, las comunicaciones entre consumidor y proveedor son punto-a-punto: del consumidor al proveedor P_i. Estrictamente hablando, para conseguir la confidencialidad deberíamos cifrar el contrato M_i, para prevenir que una tercera parte pueda monitorizar el canal de comunicaciones, y obtener su contenido. Pero esto puede evitarse utilizando algún tipo de protocolo, como Secure Socket Layer (SSL). Por lo tanto, podemos afirmar que el protocolo cumple con el requisito de confidencialidad.

VII. Conclusiones

En este artículo hemos presentado una modificación sobre un protocolo optimista para la firma electrónica de contratos en escenarios Multi-Two-Party Atómicos, que permite fusionar las fases de negociación y firma. La modificación presentada cumple con los mismos requisitos de seguridad que el protocolo original: efectividad, equitatividad, temporalidad, no-repudio y verificabilidad de la TTP. Finalmente, hemos presentado un escenario práctico en el cual podría aplicarse nuestra solución de *smart-shopping*.

Referencias

[1] G. Draper-Gil, J.-L. Ferrer-Gomila, M. F. Hinarejos, J. A. Onieva, and J. López, "Un protocolo para la firma de contratos en escenarios multi-two-party con atomicidad," in *XIIIa Reunión Española Sobre Criptología y Seguridad de la Información (RECSI XIII)*, (Arrasate, Mondragon, ES), pp. 357–362, Servicio Editorial de Mondragon Unibertsitatea, 2012.

[2] U. S. T. Committee, "Universal description, discovery and integration v3.0.2 (uddi)." http://uddi.org/pubs/uddi-v3.0.2-20041019.htm, 10 2004.

[3] N. Asokan, V. Shoup, and M. Waidner, "Optimistic fair exchange of digital signatures," in *Advances in Cryptology - EUROCRYPT'98*, vol. 1403 of *Lecture Notes in Computer Science*, pp. 591 – 606, Springer Berlin / Heidelberg, 1998.

[4] J. Zhou, R. Deng, and F. Bao, "Some remarks on a fair exchange protocol," in *Public Key Cryptography*, vol. 1751 of *Lecture Notes in Computer Science*, pp. 46 – 57, Springer Berlin / Heidelberg, 2000.

[5] Y. Liu, "An optimistic fair protocol for aggregate exchange," in *Proceedings of the 2009 Second International Conference on Future Information Technology and Management Engineering*, FITME'09, (Los Alamitos, CA, USA), pp. 564–567, IEEE Computer Society, 2009.

[6] J. A. Onieva, J. Zhou, M. Carbonell, and J. Lopez, "A multi-party non-repudiation protocol for exchange of different messages," in *18th IFIP International Information Security Conference. Security and Privacy in the Age of Uncertainty (IFIP SEC'03)*, IFIP Conference Proceedings, pp. 37–48, IFIP, 2003.

Seguridad de la información

Análisis de Riesgos Dinámico aplicado a Sistemas de Respuesta Automática frente a Intrusiones

Diego Ray
Departamento de Sistemas Telemáticos
Politécnica de Madrid
diego.raysan@alumnos.upm.es

Victor A. Villagrá
Departamento de Sistemas Telemáticos
Politécnica de Madrid
villagra@dit.upm.es

Verónica Mateos
Departamento de Sistemas Telemáticos
Politécnica de Madrid
vmateos@dit.upm.es

Pilar Holgado
Departamento de Sistemas Telemáticos
Politécnica de Madrid
pilarholgado@dit.upm.es

***Resumen*—Los Sistemas de Respuesta Automática frente a Intrusiones plantean un objetivo de interés en el campo de la Seguridad en la Información, pero desde el momento en que su fin último es asumir responsabilidades del usuario administrador, las diferentes propuestas acaban proliferando en multitud de alternativas y complejas metodologías. Las principales soluciones proponen inferir respuestas en base al estado y sucesos de red. Un administrador experto tomaría sus decisiones a raíz de estos mismos criterios, pero viéndose fuertemente influenciado por el resto de eventualidades de su entorno, tanto técnicas como corporativas, aspectos que actualmente no se contemplan o se contemplan de forma insuficiente. Apoyarse en el campo del Análisis de Riesgos como técnica madura, probada y bien regulada conduciría a estos sistemas a una mayor capacidad de respuesta y evaluación de sus acciones.**

***Palabras clave*—Análisis de Riesgos Dinámicos *(Dynamic Risk Assessment)*, Ontologías *(Ontologies)*, Sistemas de Respuestas Automáticos ante Intrusiones *(Automatic Intrusion Response Systems)*.**

I. Introducción

Los Sistemas de Respuesta Automática frente a Intrusiones (AIRS) son dispositivos de protección. Su fin último es asistir a un administrador y asumir parte de sus actuaciones, recopilando la información de su entorno y dando respuesta a anomalías y ataques. Surgen de la rama de los equipos de seguridad en redes, como cortafuegos e IDSs, destacándose por su capacidad de respuesta automática sin asistencia del administrador.

El Análisis de Riesgos (AARR) en los sistemas de información es un estudio preventivo de lo negativo que puede llegar a ocurrir. Es un estudio extenso que proporciona las guías de actuación y buenas prácticas para la protección de la información y servicios. Es un análisis a todos los niveles, y no sólo desde el punto de vista de los sistemas informáticos pero sí dirigido a ellos. Referente a este proceso integrador se resalta el apartado séptimo de la política de seguridad en la información del Ministerio de Defensa [1]: personas, documentos, SI y telecomunicaciones e instalaciones.

El alcance del artículo es dar una primera aproximación al problema y un estudio de su relevancia, estando en vías de investigación dar soluciones para un futuro prototipo. Con este fin hacer uso del estudio extenso y metodológico del riesgo para mejorar las respuestas de los AIRS. En la sección II se introducirán los conceptos de AIRS y de Análisis de Riegos Dinámicos (DRA). Se analizará el estado del arte en la aplicación del AARR en sistemas de información (SI) en general, y en los AIRS en particular. En la sección III se expondrán los principales problemas encontrados para su integración. En la sección IV los autores reflejarán su propuesta, finalizando con las conclusiones.

II. El Análisis de Riesgos en Sistemas de Respuesta frente a Intrusiones

Los AIRS se enmarcan dentro de los sistemas de defensa perimetral, servicios y tecnologías de control de acceso lógico a las redes de telecomunicación [2]. Son uno de sus elementos más recientes y se hallan actualmente en evolución. Esta familia abarca desde las implementaciones de seguridad de los IOS (*Internetwork Operating System*) de routers y switches, hasta llegar a soluciones integradas de cortafuegos, IPSs y VPNs (series ASA-5500 de CISCO). Se describen en [3] como elementos de seguridad que seleccionan y ejecutan respuestas ante las intrusiones detectadas por un IDS, proporcionando junto a [4] una posible taxonomía. De esta clasificación se extraen como características deseables: la capacidad de adaptabilidad, proactividad, sensibilidad de coste con modelo de evaluación dinámico y coherencia semántica, esta última de [5].

El AARR [6] es un área afianzada, regulada, de aplicación genérica y definida como el proceso de identificar, analizar y evaluar los riesgos [7]. Se define identificar como encontrar, reconocer y describir el riesgo, analizar como comprender y determinar su nivel de amenaza, y evaluar como determinar en que nivel es tolerable. El tratamiento específico del AARR en la seguridad en la información se encuentra regulado en normas como [8], describiéndolo como el proceso que estudia toda eventualidad que pueda afectar al activo información. El Análisis de Riesgos Dinámicos (DRA) es el propio AARR pero aplicado de forma continua, todo lo que varíe y pueda afectar a la seguridad debe de tenerse en consideración para recalcular el riesgo.

La Gestión del Riesgo (GR) consiste en analizar y tratar el riesgo [9]. El AARR identifica, analiza y evalúa activos, amenazas y salvaguardas, con los que estima el impacto (lo que podría pasar) y el riesgo (lo que probablemente pase). El tratamiento consiste fundamentalmente en aplicar las

salvaguardas para afrontar el riesgo. Derivado de lo anterior la gestión de riesgos dinámica (DMR) es consecuencia del DRA.

Los AIRS se pueden asemejar a herramientas de gestión del riesgo. Tratan el riesgo que llega al sistema y evalúan su acción de respuesta. La diferencia fundamental es que los AIRS realizan todo el proceso de forma automática, mientras que en la gestión de riesgos es el administrador el que decide las salvaguardas a aplicar.

El interés de este trabajo se centra en la aplicación de métodos de GR. Se seguirá como guía a MAGERITv3 Libro I-Método [9], que describe la gestión del riesgo como una actividad compleja, exhaustiva, en continua revisión y que requiere de una metodología probada. Partiendo de esta premisa se proponen los condicionantes para considerar la aplicación de un método de AARR en AIRS.

C1. Tratar todo activo, amenaza y salvaguarda que pueda influir en la información y servicios. "...el análisis de riesgo proporciona un modelo del sistema en términos de activos, amenazas y salvaguardas, y es la piedra angular para controlar todas las actividades con fundamento ..." [9]. Hay que establecer una relación entre los elementos del AIRS y los activos, amenazas y salvaguardas [7] del AARR. De importancia es el concepto de activo, que siguiendo la definición de [10] y priorizados según MAGERIT, se estructuran en información y servicios, como activos esenciales y que dirigen los requisitos de seguridad, y en datos, software, hardware, comunicaciones, recursos administrativos, recursos físicos y recursos humanos como activos relevantes. Atendiendo a la naturaleza de las salvaguardas los activos se clasifican en especies, encontrando en [11] una completa taxonomía por tipo de activo.

C2. Uso del riesgo y riesgo residual como producto del AARR. "...en coordinación con los objetivos, estrategia y política de la organización, las actividades de tratamiento de los riesgos permiten elaborar un plan de seguridad que, implantado y operado, satisfaga los objetivos propuestos con el nivel de riesgo que acepta la dirección. Al conjunto de estas actividades se le denomina Proceso de Gestión de Riesgos..." [9]. Se considera esta frase como fundamental, conteniendo implícitamente la finalidad de los AIRS, elementos que mantienen un nivel de seguridad dentro de un entorno. El término esencial es "nivel de riesgo aceptado", entendido como el resultado final del proceso de cálculo del riesgo ponderado con la madurez de las salvaguardas.

C3. El DRA en SI no es sólo el estudio de los cambios de red y sistemas. A lo largo del tiempo las condiciones de un entorno varían, pudiendo invalidar el análisis de riesgos vigente, punto 7.2.7 de [9]. Estas variaciones pueden deberse a nuevas amenazas, vulnerabilidades sobrevenidas, incidentes de seguridad, cambios en la utilización del sistema. De todos los expuestos los que principalmente detectan los IDSs, y sensores en general, son los incidentes de seguridad a nivel de red y sistema. Esto último separa de inmediato el significado de *dynamic risk assessment* respecto al de *online risk assessment* [12] [13] y *real time risk assessment* [14] [15], términos que en determinados contextos podrían llegar a difuminarse como se indica en [16].

C4. Integración Online/Offline. De la condición anterior se propone la separación entre un componente offline que represente a una metodología probada de AARR en el sentido de [9], y otro online que represente la aplicación del AARR en el nivel de los AIRS. Este último debe: monitorizar las incidencias de seguridad, determinar la bondad de las respuestas aplicadas y realimentar al análisis offline en el apartado de incidencias de seguridad. Los elementos online coincidirán con los que en [12] y [15] se denominan 'online' o *real-time risk assessment*. Por lo tanto el resultado del proceso offline es el riesgo y riesgo residual, del que hará uso el elemento online del AIRS.

Como conclusión se proponen las siguientes actuaciones: que se considere todo activo, amenaza y salvaguarda que afecte a los SI, que las métricas contengan al riesgo y/o riesgo residual como producto final del AARR, y que los AIRS separen sus procesos de AARR en offline y online.

II-A. DRA en Sistemas de Información

La GR en SI ya fue tratada en esta reunión por David López Cuenca y Oscar Pastor [16], por lo que esta sección se centra en su estudio desde el punto de vista del presente trabajo.

El artículo se inicia con una relación de los principales organismos nacionales e internacionales que tratan el AARR y con un glosario de la terminología del campo [8]: activos, amenazas, salvaguardas, vulnerabilidades, impacto, riesgo y riesgo residual. Continua ampliando el estudio al DRA, presentándolo como un proceso que debe de ser continuo y automático, y no aplicado a intervalos discretos como se propone en [17]. Analiza los principales trabajos sobre AARR aplicado a TI, de los que destaca [18] por contemplar el ciclo natural de un DRA. Este ciclo contiene una valoración inicial del riesgo procedente de metodologías de análisis, una realimentación proveniente de la detección de incidentes de seguridad, y por último la evaluación de las acciones tomadas con el apoyo de estas metodologías.

Prosigue presentando los principales enfoques del DRA para SI:

– *Alimentación desde BBDD* [19]. Enumera formatos estandarizados como CVE y repositorios como la *National Vulnerability Database* del NIST.
– *Grafos y árboles de ataque* [20]. Resalta el uso de redes bayesianas para el cálculo del riesgo [20] [21].
– *Monitorización del estado del sistema*. Ejemplo IDSs.

En el trabajo se señala que, ocasionalmente se tratan los términos de *real-time* y *online risk assessment* como sinónimos de *dynamic risk assessment*. Desde el punto de vista de este artículo estos términos no podrían ser sinónimos, por lo que [12] [13] [14] deberían clasificarse como métodos online, ya que si bien abarcan el estudio del AARR desde una faceta dinámica, únicamente lo hacen desde el enfoque de los incidentes de seguridad.

Derivado de ese mismo motivo, se propone que en la Fig.2 de su estudio, extraída de [18], se marque al estado –AARR Inicial– como proceso offline, y los estados –AARR

Actualizado– y –AARR Evaluar Acciones Seleccionadas– como online, cuyas métricas harían uso del estudio offline.

El enfoque que sugiere para el DRA, como actividad continua y actualizable en tiempo real, podría aplicarse únicamente al tratamiento online de los incidentes de seguridad, pero probablemente no al total de elementos que modifican el riesgo y que representan el DRA (offline). Por lo anterior se considera que se debe recurrir a ciclos discretos de revisión (PDCA [22]), o a ciclos continuos en su revisión online, cuyos resultados no se podrían reflejar instantáneamente en el cálculo del riesgo debido al complejo estudio que conllevan.

Como parte final del artículo se presenta a los AIRS como elementos de gestión y tratamiento del riesgo.

II-B. DRA en Sistemas de Respuesta ante Intrusiones

Las métricas son la herramienta usada por los AIRS para determinar y evaluar las respuestas. Es en estas métricas donde se incluyen los resultados del análisis de riesgos.

En la tesis doctoral de Mateos Lanchas [23], se proporciona una extensa comparativa entre las distintas métricas existentes, viéndose que en la mayoría de los casos, y a excepción de IDAM-IRS, no se hace uso del AARR para su cálculo. Si bien IDAM-IRS aplica términos claramente relacionados con el AARR, como *risk index* (RI) (que en términos de AARR representa el riesgo), y *risk index host* (RIH)/*risk index network* (RIN) (que en AARR representa el riesgo residual), no sigue metodologías formales para su cálculo. En el trabajo de Mateos se enuncia el uso de MAGERIT para el cálculo de métricas, pero no se destacan trabajos que lo usen.

Continuando en la línea de las recientes tesis sobre AIRS, se encuentra el trabajo de Shameli-Sendi [24]. Esfuerzo estructurado en forma de cuatro artículos que representan algunos de los principales retos que rodean a los AIRS: carácter predictivo, análisis de costos y aplicación de AARR. El primero [15] propone un marco de trabajo para la predicción de ataques multipaso, usando *Hidden Markov Models* (HMM) y lo que denomina *Alert Severity Modulation*. El segundo, *ORCEF: Online response cost evaluation framework for IRS* (estudio reflejado en la tesis pero aún no publicado como trabajo independiente), propone un IRS sensible al coste, evaluando las respuestas según dependencias entre recursos, las necesidades de QoS de los usuarios online, el daño del ataque y la confianza de la alerta. Algunas de las métricas seguidas fueron ya presentadas en el apartado 2.4 de [23]. El tercero, ARITO [25], propone la mejora de ORCEF integrando el AARR dentro de la inferencia de respuestas.

En ARITO se introducen los términos online y offline, adoptados a lo largo del presente trabajo, pero con fuertes matices diferenciados. En el proceso offline de ARITO se determina el valor del activo y lo vulnerable que es, y en el online se calcula el impacto de la alerta recibida. El proceso offline utiliza la técnica *Fuzzy Multi-Criteria Decision-Making* (FMCDM) [26], descrita en otro trabajo de los mismos autores. Para ello se determinan las propiedades de seguridad a analizar (CIA: confidencialidad, integridad y autenticidad), se identifican y clasifican los recursos, se valoran los activos según el juicio de expertos y finalmente se estudian sus vulnerabilidades.

A pesar de que ARITO parece adaptarse a nuestros requisitos, y separa entre elementos offline y online, no cumple con todos los condicionantes de la sección II. En él se propone una metodología propia para el AARR (FMCDM), que no sigue ninguna normativa o metodología, y que no deja abierta la posibilidad del uso de otros enfoques. Sin embargo, su esquema es fácilmente ampliable a otras propuestas, ya que el cálculo final lo realiza el componente online. No parece que se considere el estudio de activos más allá de la electrónica de red y sistemas, pero de nuevo esta faceta podría ampliarse incluso a través de su apuesta por FMCDM. Otro aspecto que plantea dudas es la extensión de su AARR a DRA, ya que en la fase offline necesitaría algún tipo de realimentación que influyese en la revisión del riesgo.

El mayor de los inconvenientes de ARITO es que el resultado final de su fase offline no es el riesgo, sino el valor del activo y las posibles vulnerabilidades, por lo que no se puede considerar que realiza un análisis del riesgo sino solamente una de sus etapas previas.

Sin duda los trabajos de [15] y [25] están fuertemente influenciados por DIPS (*Distributed Intrusion Prevention System*) [13]. Donde se propone un modelo distribuido activado mediante predicciones realizadas con HMM, y un análisis de riesgos realizado mediante lógica difusa (*fuzzy logic*), donde un grupo de expertos estiman el riesgo en base a una serie de variables dependientes.

Si bien DIPS es la referencia que mejor se ajusta a nuestro modelo, presenta una serie de problemas con respecto a [6] [7] y [9], que van a permitir resaltar la importancia de las condiciones propuestas en II.

En DIPS el riesgo se computa a través de reglas *fuzzy* del tipo if-then. Los parámetros del cálculo son el valor del activo, su vulnerabilidad y el nivel de amenaza. Sin embargo, en AARR el riesgo se deriva del impacto, siendo éste la degradación del valor de un activo debido a una amenaza, y de la frecuencia/probabilidad de dicha amenaza. A continuación se analizan los inconvenientes de los parámetros usados en el cálculo del riesgo en DIPS.

1. Nivel de amenaza. Se define en DIPS como la frecuencia o probabilidad, por lo que, salvando la confusión de términos [7], es un parámetro correcto en el computo del riesgo.

2. Valor del activo. En el cálculo del riesgo se debe evaluar la degradación del valor del activo (impacto). Usar únicamente el valor del activo permite determinar su importancia, pero no las consecuencias de la amenaza sobre su valor. La degradación no aparece en el cálculo con la consiguiente perdida de información en el análisis.

3. Vulnerabilidad. El riesgo residual resulta del riesgo remanente tras la aplicación de las salvaguardas. Una vulnerabilidad se define en [9] como aquello que resulta de la incapacidad de las salvaguardas, bien por su inexistencia o bien por su falta de madurez, y que impide mantener el riesgo residual aceptado. Es decir, una vulnerabilidad es algo no controlado, que si se llegase a controlar derivaría, en el peor

de los casos, en riesgo residual conocido y aceptado. Por lo que, usando estrictamente la terminología, no parece que vulnerabilidad deba formar parte del cálculo del riesgo, ya que es consecuencia de la imperfección del propio análisis, bien en la fase de identificación o bien en el modelado de las salvaguardas. Obviamente en DIPS vulnerabilidad toma otro significado, probablemente referido a la degradación sufrida en el valor del activo.

4. Activo. En DIPS se define como todo dato, equipo u otro componente que soporte a la información de la actividad. No se proporciona una clasificación formal de activos como la presentada en [10] y [9], dificultando un estudio metódico y estructurado.

Con el razonamiento anterior sobre DIPS quiere ponerse de manifiesto que, si bien la solución aportada es de las más destacadas, al no hacer uso de metodologías que se adapten a las normativas y recomendaciones vigentes, ni de una terminología regulada, se introducen conceptos ambiguos desde el punto de vista del AARR. Estas mismas ambigüedades reducen la posibilidad de adaptar los procesos online y offline que proviniesen de distintos trabajos.

III. El Problema Dimensional entre el DRA y los AIRS

El título del apartado pretende resaltar la distancia existente entre el campo del DRA y el de los AIRS. A continuación se presentan los principales retos que hacen que su interrelación resulte compleja pero conveniente.

1. Carácter Preventivo frente a Reactivo. El análisis de riesgos es un estudio preventivo que sugiere qué hacer en caso de que algo negativo ocurra. Su fin no es responder a las amenazas, sino proponer soluciones y estimar riesgos. Por el contrario, los AIRS tienen como fin actuar, y aunque necesiten analizar ciertos factores que aseguren la calidad de las respuestas, no realizan un estudio exhaustivo o integral del entorno. Como se verá, el nexo entre lo preventivo del AARR y lo reactivo de los AIRS es el concepto de nivel de confianza.

2. Actividad a Medio-Largo Plazo frente a Tiempo Real. Una de las propiedades deseables de los AIRS es su rapidez de respuesta, mientras que el análisis de riesgos es una actividad ardua, continua y cíclica, en la que prima más el buen análisis a medio-largo plazo que un reajuste rápido de la situación. Por ello el AIRS hace uso de resultados previamente calculados por los elementos offline del AARR, reajustando y orientando sus respuestas, pero asumiendo que la dinámica de trabajo del AARR es lenta.

3. Estudio Exhaustivo frente a Especializado. El análisis de riesgos es un estudio extenso, a todos los niveles, que contempla tanto factores tecnológicos, como sociales, políticos, legales y de cualquier otra índole. Por el contrario, los sistemas de respuesta generalmente realizan un estudio muy dirigido, centrándose casi en exclusiva en las eventualidades de red y sistemas. Incluso cuando se introduce el concepto de usuario, se hace desde el punto de vista de 'persona que usa el sistema', y no como 'persona que puede causar un daño', daño que no tiene que ser técnico, pero que sí puede repercutir gravemente en la información y los servicios.

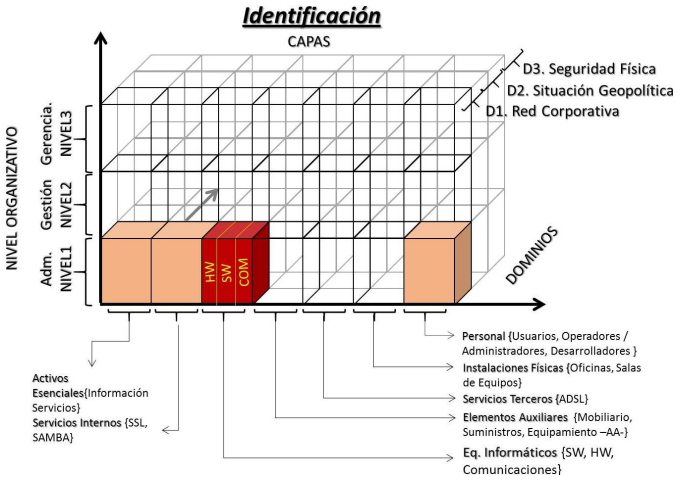


Figura 1. Identificación de Activos

La identificación de activos, amenazas y salvaguardas es la primera fase del AARR. En la figura 1 se presenta la determinación de estos activos, los ejes de dominios y capas corresponden con lo recomendado en [9], y nivel organizativo es una dimensión añadida en este trabajo. A continuación se detallan.

– *Capas.* Atendiendo al tipo de activo y sus dependencia jerárquicas. Se suele dividir en subcapas, la capa equipos informáticos contiene a las subcapas hw, sw y comunicaciones.

– *Dominios.* Por agrupación de amenazas y salvaguardas que afectan al activo. Como ejemplo se seleccionan los dominios de seguridad física, red corporativa, y situación geopolítica.

– *Nivel organizativo.* Permite analizar el activo de una determinada capa-dominio desde diferentes niveles de decisión. En este caso se contemplan los niveles administración, gestión y gerencia.

La intención de la figura 1 es resaltar la información que se omite en el caso de no aplicar el AARR como metodología. De todas las dimensiones (capa/dominio/nivel) reflejadas en la figura 1, los únicos activos generalmente analizados por los AIRS son los de la capa de equipos informáticos, y únicamente en el nivel de administración de red y dentro del dominio de la red corporativa (cubo resaltado)

IV. Enfoque para la Integración del Análisis de Riesgos

En este apartado se presentan las bases de una propuesta temprana. Se explican sus principales conceptos y se propone el AIRS que servirá de base para la integración.

Bases de la Propuesta

Se propone un marco de actuación basado en el conocimiento, con el objetivo de mantener un nivel de confianza dentro de un dominio de control, dado un estado de alerta, en beneficio

de una entidad objetivo y centrado en sus tecnologías de la información, pero atendiendo a toda eventualidad.

El núcleo del marco lo conforma el concepto de sistema de respuesta automática ante intrusiones, como elemento de respuesta, evaluación y análisis de riesgos integral.

Conceptos Principales

– *Basado en el Conocimiento.* Se entenderá esta propiedad como la capacidad del sistema de razonar conforme a conceptos y no sobre sintaxis. Término equivalente a la propiedad de coherencia semántica del AIRS ontológico de Mateos Lanchas [27]. Las definiciones necesarias para la integración se realizarán mediante ontologías en OWL2 [28], reglas SWRL [29] y razonadores como Pellet [30].

– *Mantener un Nivel de Confianza.* Es el núcleo de la propuesta, y el elemento que pretende conferir al sistema un carácter convergente hacia un estado deseado. En [9] se define seguridad como la capacidad de resistir, manteniendo un determinado nivel de confianza. Este nivel se medirá en diferentes dimensiones de seguridad: confidencialidad, integridad, autenticidad, trazabilidad y disponibilidad, y para cada activo, amenaza y dimensión.

El elemento de AARR que se usará como métrica del nivel de confianza es el riesgo residual o riesgo aceptado, como producto final de las metodologías de análisis de riesgos. C2 sección II.

A continuación se justifica la elección del riesgo residual. El fin del AARR es calcular el riesgo, computado a partir del impacto que una amenaza tiene sobre la degradación del valor de los activos y ponderado por su probabilidad o frecuencia. El riesgo residual es el riesgo aceptado y conocido, residuo remanente una vez aplicadas las salvaguardas, y que determina el rango de riesgo que una organización está dispuesta o condicionada a admitir. Por lo anterior, el riesgo residual es el término que contiene la totalidad de la información del proceso de AARR, tanto de la valoración del riesgo, como de la eficacia de las salvaguardas. Como nota final, y aunque implícito en el propio cálculo del riesgo, se introduce el término de riesgo residual dimensional, consistente en el riesgo residual medido en cada una de las dimensiones de seguridad seleccionadas.

El nivel de confianza debe de representarse tanto en los componentes offline como en los online, C3/4 de IV, para ello se introducen los siguientes conceptos:

– *Nivel de Confianza Objetivo.* Representa el resultado del proceso de AARR offline, caracterizado por el riesgo residual. El AIRS debe converger a este nivel de confianza, utilizando sus valores en el cálculo de las métricas. Un cambio en el nivel de confianza conlleva un cambio en las respuestas seleccionadas, y en ocasiones el modelado de nuevas salvaguardas/respuestas.

– *Nivel de Confianza Actual.* Representa el resultado del proceso de AARR online, caracterizado por el riesgo residual evaluado online, término introducido en este estudio y no propio del análisis de riesgos. Este riesgo evaluado debe de medirse en toda dimensión de todo activo afectado por un ataque (amenaza llevada a cabo), para poder compararlo con el nivel de confianza objetivo de la fase offline. Tras la aplicación de las salvaguardas en forma de repuestas, el riesgo residual evaluado debe de estar en los márgenes definidos por el riesgo residual objetivo. De no ser así, las salvaguardas han sido insuficientes, debiendo el componente online realimentar al offline con esta información y replantearse un nuevo ciclo de AARR en su faceta dinámica. Como se explicó, la realimentación del componente online afectaría principalmente a las incidencias de seguridad dentro del DRA.

– *Estado de Alerta.* Si mantener el nivel de confianza provoca la convergencia hacia una situación estable, el estado de alerta juega el papel de elemento regulador. El estado de alerta se valora, en términos de análisis de riesgos, según el grado de impacto con que las amenazas afectan a la organización. Un determinado problema puede tener mayores o menores consecuencias dependiendo de la situación, y esto debe de reflejarse en el estado de alerta e influir en las respuestas del AIRS. El estado de alerta podrá: imponerse por el administrador, derivarse de históricos que relacionen situaciones anteriores y/o derivarse de la tendencia del impacto a lo largo de un periodo.

El estado de alerta medido según la tendencia del impacto abre la posibilidad de realizar un estudio preventivo de incidencias futuras. Los AIRS pueden hacer uso de esta característica en su faceta proactiva en el caso de implementarla. Se propone el uso del *gradiente* para el cálculo de esta tendencia, donde los vectores unitarios *i,j,k* ... representan a las dimensiones de seguridad, y el valor de cada eje la tendencia del impacto.

Elección del AIRS

Los primeros estudios se realizarán en base al AIRS ontológico propuesto en [27] y exhaustivamente descrito en [23]. Para ello se prevé como mínimo:

– *Módulo de Análisis de Riesgos Dinámicos.* Será parte del modelo online y realizará la función de adaptar los resultados del AARR offline a nuestro sistema. Mantendrá los mismos criterios de coherencia semántica que para la recepción de alertas del AIRS ontológico.

– *Ampliación del Módulo de Evaluaciones.* Determinará el nivel de confianza actual, por lo que sus resultados, en términos de evaluación de riesgo residual, marcarán la capacidad de convergencia al nivel de confianza objetivo. Entre los elementos previstos se encuentra la especificación de un toolkit de evaluaciones, similar y en estrecha relación con en toolkit de respuestas ya existente en [27].

V. Conclusión

Los AIRS se plantean como una potente herramienta en la administración de la seguridad en redes y sistemas. La calidad de sus respuestas depende de un gran número de factores, pero principalmente de la fiabilidad de los eventos recibidos, de las métricas que los analizan e infieren las respuesta, y de los mecanismos de evaluación de las acciones tomadas. La búsqueda de nuevas propiedades como la proactividad [31], y

la mejora de los criterios de respuesta mediante el AARR son algunos de los retos actuales dentro del campo.

Ajustándose a los términos que este artículo propone, se puede clasificar a los AIRS como herramientas de apoyo a la gestión del riesgo, que ***reciben*** información del proceso de AARR para la inferencia de sus respuestas y ***proporcionan*** información a los procesos de DRA para el reajuste del riesgo.

Como aportaciones del trabajo se propone desambiguar el término *dynamic risk assessment* con respecto a *online* o *real time risk assessment*. Se propone una distinción y separación entre los componentes offline y online, el primero como AARR propiamente y el segundo como proceso que hace uso de los resultado del offline y lo realimenta. Se presentan las principales diferencias entre la naturaleza del AARR y AIRS: preventivo/reactivo, medio-plazo/tiempo-real, estudio-extenso/especializado. Finalmente se dan las líneas de un posible mecanismo para la aplicación del AARR en los AIRS, con base en el AIRS ontológico de [27].

La integración propuesta se presenta en forma de ideas fuerza que derivan en conceptos que relacionan e intentan salvar las diferencias existentes entre el DRA y los AIRS. Se propone al ***riesgo residual*** como elemento que defina el ***nivel de confianza*** de una organización, identificándolo como el producto final del AARR y el contenedor de mayor información. El riesgo y el riesgo residual serían el resultado del componente offline. Los procesos online harán uso de este riesgo calculado en sus respuestas y evaluarán el riesgo residual provocado por las incidencias de seguridad, siendo éste el nivel de confianza existente en un momento dado. Con este riesgo online se realimentará a los procesos offline en su faceta dinámica.

Referencias

[1] M. de Defensa, "Om-76/2006, política de seguridad en la información del ministerio de defensa," p. Séptimo. Áreas de la seguridad de la información, 2006.

[2] V. A. Villagrá, *Seguridad en Redes de Telecomunicación*. Fundación Rogelio Segovia para el Desarrollo de las Telecomunicaciones, 2009.

[3] N. Stakhanova, S. Basu, and J. Wong, "A taxonomy of intrusion response systems," *Int.J.Inf.Comput.Secur.*, vol. 1, no. 1/2, pp. 169–184, jan 2007.

[4] A. Shameli-Sendi, N. Ezzati-Jivan, M. Jabbarifar, and M. Dagenais, "Intrusion response systems: survey and taxonomy," *Internacional Journal of Computer Science and Network Security (IJCSNS)*, vol. 12, no. 1, pp. 1–14, 2012.

[5] V. Mateos, V. A. Villagrá, F. R. Bueno, and J. Berrocal, "Definition of response metrics for an ontology-based automated intrusion response systems," *Computers & Electrical Engineering*, vol. 38, no. 5, pp. 1102–1114, 2012.

[6] ISO-31000:2009, *Risk management - Principles and guidelines*. International Organization for Standardization, ISO, 2009.

[7] ISO-Guide-73:2009, *Risk management and Vocabulary*. International Organization for Standardization, ISO, 2009.

[8] ISO/IEC-27005:2011, *Information technology, security techniques, Information security risk management*. International Organization for Standardization, ISOIEC, 2011.

[9] M. A. Amutio, J. Candau, and J. A. Mañas, *MAGERITv3, Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. Libro I - Método*. Ministerio de Hacienda y Administraciones Públicas, 2012.

[10] UNE-71504:2008, *Metodología de análisis de riesgos para los sistemas de información*. AENOR, 2008.

[11] M. A. Amutio, J. Candau, and J. A. Mañas, *MAGERITv3, Metodología de Análisis y Gestión de Riesgos de los Sistemas de Información. Libro II - Catálogo de Elementos*. Ministerio de Hacienda y Administraciones Públicas, 2012.

[12] C. P. Mu, X. J. Li, H. K. Huang, and S. F. Tian, "Online risk assessment of intrusion scenarios using d-s evidence theory," in *European Symposium on Research in Computer Security (ESORICS)*, 2008, pp. 35–48.

[13] K. Haslum, A. Abraham, and S. J. Knapskog, "Dips: A framework for distributed intrusion prediction and prevention using hidden markov models and online fuzzy risk assessment," in *Third International Information Assurance and Security Symposium (IAS)*, 2007, pp. 183–190.

[14] Z.-H. Hu, Y.-S. Ding, and J.-W. Huang, "Knowledge-based framework for real-time risk assessment of information security inspired by danger model," in *Proceedings of the 2008 International Symposium on Intelligent Information Technology Application Workshops (IITAW '08)*, ser. IITAW '08, 2008, pp. 1053–1056.

[15] A. Shameli-Sendi, M. Dagenais, M. Jabbarifar, and M. Couture, "Real time intrusion prediction based on optimized alerts with hidden markov model," *Journal of Networks (JNW)*, vol. 7, no. 2, pp. 311–321, 2012.

[16] D. L. Cuenca, O. Pastor, and L. J. Villalba, "Concepto y enfoques sobre el análisis y la gestión dinámica del riesgo en sistemas de información," in *Reunión Española sobre Criptología y Seguridad de la Información (RECSI)*, 2012.

[17] W. Qi, X. Liu, J. Zhang, and W. Yuan, "Dynamic assessment and var-based quantification of information security risk," in *2nd International e-Business and Information System Security Conference (EBISS),*, 2010, pp. 1–4.

[18] P. Lagadec, "Visualization et analyse de risque dynamique pour la cyber-défense," in *Symposium sur la sécurité des technologies de l?information et des communications (SSTIC)*, 2010.

[19] W. Jones, S. Aud, J. Hudepohl, M. flournory, W. Snipes, and E. Schutz, "Method and system for dynamic risk assessment of software," 2001.

[20] J. A. Mañas and C. Belso, "Gestión dinámica de riesgos: Seguridad de la red de servicios," in *XI jornadas sobre tecnologías de la información para la modernización de las administraciones públicas*, 2010.

[21] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using bayesian attack graphs," *Dependable and Secure Computing, IEEE Transactions on*, vol. 9, no. 1, pp. 61–74, 2012.

[22] ISO/IEC-27001:2013, *Information technology, security techniques, Information security managament systems, Requirements*. International Organization for Standardization, ISOIEC, 2013.

[23] V. Mateos, "Contribución a la automatización de sistemas de respuesta frente a intrusiones mediante ontologías," 2013. [Online]. Available: http://www.dit.upm.es/~doct/ist/tesisleidas.html

[24] A. Shameli-Sendi, "System health monitoring and proactive response activation," 2013. [Online]. Available: http://publications.polymtl.ca/1102/1/2013_AlirezaShameliSendi.pdf

[25] A. Shameli-Sendi and M. Dagenais, "Arito: Cyber-attack response system using accurate risk impact tolerance," *International Journal of Information Security*, pp. 1–24, 2013.

[26] A. Shameli-Sendi, M. Shajari, M. Hassanabadi, M. Jabbarifar, and M. Dagenais, "Fuzzy multi-criteria decision-making for information security risk assessment," *The Open Cybernetics and Systemics Journal*, vol. 6, pp. 26–37, 2012.

[27] V. Mateos, V. A. Villagrá, and F. Romeo, "Ontologies-based automated intrusion response system," in *Proceedings of the 3rd international conference on computational intelligence in security for information systems (CISIS)*, 2010, pp. 99–106.

[28] D. L. Michael K. Smith, Chris Welty, *OWL 2 Web Ontology Language Document Overview*, 2nd ed., December 2012. [Online]. Available: http://www.w3.org/TR/2012/REC-owl2-overview-20121211/

[29] P. F. P.-S. Ian Horrocks, *SWRL: A Semantic Web Rule Language Combining OWL and RuleML*, May 2004. [Online]. Available: http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

[30] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, "Pellet: A practical owl-dl reasoner," *Web Semant.*, vol. 5, no. 2, pp. 51–53, jun 2007.

[31] N. Stakhanova, S. Basu, and J. Wong, "A cost-sensitive model for preemptive intrusion response systems." in *Proceedings of the 21st international conference on advanced networking and applications AINA'07*. IEEE Computer Society, 2007, pp. 428–435.

Simulación de la propagación del malware: Modelos continuos vs. modelos discretos

Amparo Fúster Sabater
Instituto de Tecnologías Físicas y de la Información, C.S.I.C.
Email: amparo@iec.csic.es

Ángel Martín del Rey
Departamento de Matemática Aplicada
IUFFyM, Universidad de Salamanca
Email: delrey@usal.es

Gerardo Rodríguez Sánchez
Departamento de Matemática Aplicada
IUFFyM, Universidad de Salamanca
Email: gerardo@usal.es

***Resumen*—La gran mayoría de modelos matemáticos propuestos hasta la fecha para simular la propagación del malware están basados en el uso de ecuaciones diferenciales. Dichos modelos son analizados de manera crítica en este trabajo, determinando las principales deficiencias que presentan y planteando distintas alternativas para su subsanación. En este sentido, se estudia el uso de los autómatas celulares como nuevo paradigma en el que basar los modelos epidemiológicos, proponiendo una alternativa explícita basada en ellos a un reciente modelo continuo.**

***Palabras clave*—Autómatas Celulares (*cellular automata*). Código malicioso (*malware*). Ecuaciones diferenciales (*differential equations*). Epidemiología matemática (*mathematical epidemiology*). Modelización matemática (*mathematical modelling*).**

I. Introducción

El malware es una de las principales amenazas a la Seguridad de la Información con la que nos enfrentamos en la actualidad. Esta amenaza (y los efectos causados), lejos de disminuir, se acrecentará en los próximos años debido fundamentalmente al perfeccionamiento de sus técnicas y fines (APT, Crimeware, etc.) y a la progresiva implantación de la Internet de las Cosas. La lucha contra el malware se lleva a cabo en diferentes frentes: desde la concienciación del usuario para que adopte medidas de seguridad, hasta el desarrollo de software antimalware por parte de las empresas especializadas, pasando por el establecimiento de políticas de seguridad adecuadas en los distintos organismos y compañías, etc. El gran olvidado en este escenario es el desarrollo de software simulador de la propagación de malware. Este tipo de aplicaciones, tan usadas en otros campos como en la propagación de enfermedades infecciosas o de incendios forestales, sería de gran utilidad para el gestor ya que le permitiría simular el comportamiento de la propagación del código malicioso en una red, probar la efectividad de contramedidas y, en definitiva, tomar decisiones adecuadas para la contención de la propagación o, al menos, la minimización de sus efectos nocivos. El software de simulación se deriva de la implementación computacional de un determinado modelo matemático. Así pues el desarrollo de este tipo de modelos que traten de explicar el comportamiento de la propagación del código malicioso es básico.

Existen muy pocos modelos publicados en la literatura científica cuyo propósito sea el mencionado anteriormente; la gran mayoría de ellos se basan en el paradigma heredado de la Epidemiología Matemática y más concretamente en el modelo de Kermack y McKendrick ([5]) que hace uso de un sistema de ecuaciones diferenciales. Aunque estos modelos continuos poseen una sólida base matemática que posibilita un estudio cualitativo muy detallado, presentan serios problemas a la hora de aplicarlos en determinadas situaciones reales. Ello hace conveniente explorar otro tipo de herramientas matemáticas de naturaleza discreta (autómatas celulares, modelos basados en agentes, etc.) que posibiliten el diseño de modelos más eficaces.

El objetivo fundamental de este trabajo es realizar un análisis crítico de los modelos matemáticos propuestos, determinando los puntos fuertes y débiles y, a partir de ello, proponer paradigmas alternativos que permitan soslayar los problemas planteados por los existentes. En este sentido, y con la finalidad de ilustrar las conclusiones obtenidas, se propone un modelo basado en autómatas celulares para el estudio de la propagación de malware, alternativo al desarrollado por Feng *et al.* ([3]) que se basa en ecuaciones diferenciales ordinarias.

El resto del trabajo está organizado como sigue: en la sección II se analizan los modelos matemáticos que se han desarrollado para simular la propagación del malware, determinando ventajas y desventajas; en la sección III se detalla el modelo continuo debido a Feng *et al.*. La propuesta y análisis de la alternativa discreta es presentada en la sección IV; finalmente, en la sección V se presentan las conclusiones.

II. Modelos matemáticos basados en ecuaciones diferenciales

Los modelos matemáticos desarrollados para estudiar la propagación de malware se basan en los modelos diseñados para estudiar la diseminación de las enfermedades infecciosas; ello es debido a las similitudes entre el comportamiento de los virus biológicos, bacterias, hongos o priones y el del malware (virus computacionales, gusanos, etc.) Así pues muchas de las propiedades y características de los primeros se traducen y tienen su reflejo en los segundos (véase [10], [14]), a saber: clases en que se divide la población, la naturaleza del modelo y mecanismos que rigen la dinámica de la infección. Los modelos epidemiológicos de carácter matemático son modelos compartimentales, esto es, la población se divide en diferentes tipos (o compartimentos) teniendo en cuenta las características de la enfermedad: susceptibles, expuestos (con

o sin síntomas), infectados, infecciosos, recuperados, en cuarentena, vacunados, aislados, etc. Así pues nos podemos encontrar con modelos SIS (Susceptible-Infectado-Susceptible), modelos SIR (Susceptible-Infectado-Recuperado), modelos SEIR (Susceptible-Expuesto-Infectado-Recuperado), modelos SEIQR (Susceptible-Expuesto-Infectado-Cuarentena-Recuperado), etc.

Consecuentemente, en los modelos cuyo objeto de estudio es el malware podemos encontrar estos mismos compartimentos; así se han propuesto modelos SIS (véase, por ejemplo, [1]), SIR (véase, por ejemplo, [13]), SEIR (véase, por ejemplo, [12]), SEIRS (véase [15]), etc. Se puede ver cómo no existe un tipo de modelo compartimental que centre el mayor número de trabajos sino que se observa una cierta homogeneidad en cuanto a los modelos compartimentales propuestos. Estos modelos se pueden clasificar también atendiendo a la naturaleza y a las herramientas matemáticas en las que se basan. En este sentido nos podemos encontrar con modelos deterministas (véase, por ejemplo [11], [17]) o con modelos estocásticos ([2], [6]). Los modelos deterministas están basados en ecuaciones diferenciales, mientras que los modelos estocásticos hacen uso fundamentalmente de las cadenas de Markov (sobre tiempo y estados continuos o discretos). Los modelos deterministas proporcionan buenos resultados cuando la población es muy grande, mientras que los modelos estocásticos se muestran más eficaces cuando se intenta simular la propagación de malware en redes pequeñas de ordenadores. La gran mayoría de los modelos propuestos (ya sean deterministas o estocásticos) se pueden calificar como modelos globales ya que estudian la dinámica de la población en su conjunto sin tener en cuenta las interacciones locales entre los individuos más allá de lo reflejado en los parámetros. Por el contrario existen muy pocos modelos de carácter individual; todos ellos basados en autómatas celulares (véase [4], [7], [8], [9]).

El objetivo de la inmensa mayoría de los modelos propuestos es el estudio de la dinámica de los diferentes compartimentos en que se divide la población, es decir, el conocimiento del número de ordenadores susceptibles, expuestos, infectados, etc. que hay en cada instante de tiempo y cuál es su tendencia.

Todo modelo matemático viene caracterizado por tres elementos: las variables que se estudian, los parámetros que se utilizan y las relaciones funcionales que rigen la dinámica considerando las variables y parámetros. En el caso de la simulación de la propagación del malware, las variables utilizadas son el número de ordenadores que se encuentran en alguno de los tipos considerados. Los parámetros que se utilizan en la modelización suelen ser los siguientes (el uso de unos u otros depende del modelo implementado y del tipo de malware considerado): tasa de infección, tasa de recuperación (debida al efecto de los antivirus), índice de eliminación de un ordenador de la red, índice de aparición de nuevos ordenadores en la red, probabilidades de paso de un compartimento a otro, probabilidad de adquisición de inmunidad (temporal o indefinida), periodo de latencia, periodo de inmunidad, etc. La evolución de los diferentes compartimentos viene regida por las relaciones funcionales que tienen en cuenta los parámetros introducidos en el modelo. Estas relaciones se pueden articular en torno a diferentes herramientas matemáticas, siendo la más utilizada las ecuaciones diferenciales.

El pilar sobre el que se fundamentan los modelos basados en ecuaciones diferenciales es el modelo de Kermack y McKendrick ([5]). Se trata de un modelo SIR en el que el tamaño de la población se mantiene constante e igual a N y se consideran dos parámetros: el índice de transmisión a, y el índice de recuperación b. La dinámica del modelo se rige según el siguiente sistema de ecuaciones diferenciales ordinarias:

$$\begin{cases} S'(t) = -\frac{a}{N} S(t) I(t) \\ I'(t) = \frac{a}{N} S(t) I(t) \\ R'(t) = bI(t) \end{cases} \tag{1}$$

donde $S(t), I(t)$ y $R(t)$ representan el número de ordenadores susceptibles, infectados y recuperados en el instante t, respectivamente.

El uso de ecuaciones diferenciales permite realizar un detallado análisis matemático del modelo en cuestión. El comportamiento de estos modelos depende fundamentalmente de un parámetro umbral llamado número reproductivo básico, R_0, el cual determinará la estabilidad del equilibrio sin infección (*disease-free equilibrium*) y del equilibrio endémico (*endemic equilibrium*). El número reproductivo básico se define como el número de infecciones secundarias causadas por un único ordenador infectado en una población enteramente susceptible. De esta manera, se demuestra que si $R_0 < 1$ la infección se irá reduciendo (el número de ordenadores infectados decrecerá hasta erradicarse) alcanzándose un estado de equilibrio estable sin infección; si, por el contrario, se verifica que $R_0 > 1$, entonces la infección se propagará (el número de individuos infectados crecerá) llegándose a un estado de equilibrio endémico estable.

Se trata pues de modelos bien fundamentados y coherentes desde el punto de vista matemático y con un detallado estudio de las principales características de su dinámica: estabilidad, equilibrio, etc. No obstante presentan algunos inconvenientes que pasaremos a detallar a continuación y que son debidos a su propia naturaleza:

(1) No tienen en cuenta las interacciones locales entre los ordenadores que forman la red. Se utilizan parámetros como la tasa de infección, la tasa de recuperación, etc. que son de carácter general: el valor del parámetro es constante para todos los elementos de la red o, en algunos casos, sigue una determinada distribución de probabilidad. Consecuentemente no se contempla el uso de parámetros individualizados para cada uno de los ordenadores.

(2) Suponen que los ordenadores que forman la red (a través de la que se propaga el código malicioso) están homogéneamente distribuidos y conectados todos entre sí. Cuando se analiza la propagación del código malicioso de manera macroscópica (en toda Internet, por ejemplo) los resultados que se obtienen dan una aproximación bastante buena de lo que ocurre en la realidad; ahora bien, si analizamos dicha propagación en redes locales,

intranets, etc. los resultados obtenidos son manifiestamente mejorables ya que a escala microscópica la dinámica es muy sensible a las interconexiones locales.

(3) No es posible simular la dinámica individual de cada uno de los elementos de la red. Bien es cierto que, cuando el tamaño de la red es muy grande, el comportamiento general obtenido puede ser muy similar (en cuanto a tendencias) a lo que se produce en la realidad pero se omite el uso de información fundamental: por ejemplo aquellas computadoras cuyo sistema operativo sea Mac OS no se deberían ver afectadas (en el sentido de ser infectadas) por el código malicioso diseñado para sistemas que utilicen Windows (aunque podrían considerarse expuestas), etc.

Consecuentemente, en los modelos basados en ecuaciones diferenciales podemos obtener buenos resultados acerca del comportamiento global aunque no tendremos información sobre el comportamiento individual de cada una de los ordenadores de la red. Estas tres deficiencias fundamentales que presentan estos modelos podrían ser subsanadas si utilizáramos otro tipo de modelos como los basados en autómatas celulares. En éstos es posible tener en cuenta las características individuales de cada una de las computadoras o dispositivos que se encontraran conectados a la red; además podríamos considerar diferentes topologías de red e incluso variarlas con el tiempo. De esta manera tendríamos definido un modelo en el que la dinámica variara en función de los distintos parámetros individuales.

III. El modelo de Feng *et al.*

En [3] Feng *et al.* propusieron un modelo SIRS basado en un sistema de ecuaciones diferenciales ordinarias con retardo para simular la propagación de un determinado código malicioso. Este modelo se caracteriza porque en él se considera una tasa de infección $\beta(t)$ variable, un cierto periodo de inmunidad temporal τ tras la eliminación satisfactoria del malware, y se supone que el número total de ordenadores puede variar con el tiempo: $S(t)+I(t)+R(t)=N(t)$.

La dinámica del mismo viene definida por las siguientes consideraciones:

(1) Un ordenador susceptible pasa a estar infectado con tasa de infección $\beta(t)$. Este índice depende de múltiples factores como: número de ordenadores susceptibles, daños causados por el malware, etc.

(2) Un ordenador susceptible (*resp.* infectado) pasa a estar recuperado con tasa de inmunidad ϕ (*resp.* γ) si sobre él se tienen implementadas diferentes medidas de seguridad: software antivirus, firewall, sistema de detección de intrusos, etc.

(3) Un ordenador recuperado pasa a ser susceptible según la tasa δ después de un cierto periodo de tiempo τ.

Concretamente, las ecuaciones que rigen el modelo son las siguientes:

$$\begin{aligned} S'(t) &= p\Lambda-\beta(t)S(t)I(t)-(\mu+\phi)S(t) \\ &\quad +\delta R(t-\tau) \\ I'(t) &= \beta(t)S(t)I(t)-(\mu+\gamma)I(t) \\ R'(t) &= (1-p)\Lambda+\phi S(t)+\gamma I(t)-\delta R(t-\tau) \\ &\quad -\mu R(t). \end{aligned} \tag{2}$$

Tabla I: Parámetros del modelo debido a Feng *et al.*

Parámetro	Descripción
p	Porcentaje de ordenadores susceptibles
Λ	Número de nuevos nodos
δ	Tasa de pérdida de inmunidad
$\beta(t)$	Tasa de infección en el instante t
μ	Tasa de reposición de ordenadores
ϕ	Tasa de inmunidad proporcionada por el software antivirus
γ	Tasa de recuperación de la infección

de manera que los parámetros utilizados se muestran en la tabla I.

Obsérvese que éstos son parámetros globales, es decir, el valor de cada uno de ellos es constante sobre todos los ordenadores de la red.

Un laborioso cálculo matemático (véase [3]) demuestra que el número reproductivo básico asociado es:

$$R_0=\frac{\beta_0(p\mu+\delta)\Lambda f'(0)}{\mu(\mu+\gamma)(\mu+\delta+\phi)}, \tag{3}$$

donde $f(t)=\frac{\beta(t)I(t)}{\beta_0}$, siendo β_0 la tasa inicial de infección. Además, si $R_0\leq 1$ se obtiene el estado de equilibrio sin infección dado por $E_0^*=(S_0^*,I_0^*,R_0^*)$, donde:

$$S_0^*=\frac{(p\mu+\delta)\Lambda}{\mu(\mu+\delta+\phi)}, I_0^*=0, R_0^*=\frac{(1-p)\Lambda+\phi S_0^*}{\delta+\mu}. \tag{4}$$

Se verifica que E_0^* es globalmente asintóticamente estable para cualquier τ si $R_0<1$. Por otro lado, si $R_0>1$ entonces se alcanza un estado de equilibrio endémico. Se demuestra que dicho estado es localmente asintóticamente estable si $\tau<\tau_0$ e inestable cuando $\tau>\tau_0$, donde τ_0 es un cierto parámetro umbral.

IV. El modelo basado en autómatas celulares

IV-A. *Descripción del modelo*

En el modelo de Feng *et al.* descrito en la Sección III se emplean parámetros generales sin atender a las características específicas de cada uno de los ordenadores que se encuentra en la red ni a las posibles conexiones entre ellos. A fín de tener en cuenta estos condicionantes, proponemos un modelo alternativo basado en autómatas celulares cuyos resultados globales son los similares a los obtenidos por el modelo original pero que, al mismo tiempo, permite incorporar las características particulares de cada uno de los ordenadores y obtener, adicionalmente, la evolución temporal de los mismos.

Los autómatas celulares son modelos simples de computación (un tipo particular de modelos basados en agentes) que

son capaces de simular de manera eficaz y eficiente sistemas complejos (véase [16]). Están formados por un número finito de unidades de memoria denominadas células que se encuentran conectadas entre sí según una cierta topología definida por un grafo, de tal manera que en cada instante de tiempo cada célula está en un estado de entre un número finito de ellos. Este estado va cambiando con el paso discreto del tiempo de acuerdo una regla de transición local cuyas variables son los estados en el instante anterior de la propia célula y sus vecinas (aquellas células adyacentes a la dada).

En el caso que nos ocupa supondremos que cada célula representará un ordenador de la red considerada y que la vecindad de la misma vendrá definida por el conjunto de ordenadores que se encuentran conectados de manera que sea posible la transmisión del malware entre ellos (vía correo electrónico, bluetooth, etc.) A este respecto denotaremos por $[i]$ al i-ésimo ordenador de la red y por $V_i = \{[j_{i,1}], [j_{i,2}], \ldots, [j_{i,v_i}]\}$ a su vecindad. El estado de la célula/ordenador i-ésimo en el instante de tiempo t se denotará por $E_i(t)$ y tomará alguno de los siguientes tres valores: S (susceptible), I (infectado), o R (recuperado). La transición entre dichos estados vendrá regida por las siguientes suposiciones:

(1) *Transición de susceptible a infectado*: El ordenador susceptible $[i]$ pasará estar infectado cuando exista un ordenador vecino infectado, en cuyo caso dicha infección se producirá con probabilidad $\beta_i(t)$. La existencia de un vecino infectado vendrá dada por la siguiente variable booleana:

$$r_{i,j}(t) = \begin{cases} 1, & \text{si } [j] \text{ está infectado en } t \\ 0, & \text{si } [j] \text{ no está infectado en } t \end{cases} \tag{5}$$

donde $[j] \in V_i$.

(2) *Transición de susceptible a recuperado*: El ordenador susceptible $[i]$ pasará a estar recuperado cuando se tomen las medidas necesarias para que el malware no le afecte. Ello se producirá con probabilidad $\phi_i(t)$.

(3) *Transición de infectado a recuperado*: El ordenador infectado $[i]$ pasará a estar recuperado cuando tenga software antivirus instalado, en cuyo caso la recuperación se producirá con probabilidad $\gamma_i(t)$. La existencia de software de protección vendrá dada por la siguiente variable booleana:

$$s_i(t) = \begin{cases} 1, & \text{si } [i] \text{ tiene antivirus instalado en } t \\ 0, & \text{si } [i] \text{ no tiene antivirus instalado en } t \end{cases} \tag{6}$$

(4) *Transición de recuperado a susceptible*: Un ordenador recuperado se mantendrá en este estado durando un cierto periodo de tiempo: τ_i unidades temporales discretas. Posteriormente pasará a encontrarse en estado susceptible con probabilidad $\delta_i(t)$.

Consecuentemente, las respectivas funciones de transición local serán las siguientes:

$$E_i(t+1) = \begin{cases} S & \text{si } E_i(t) = S \text{ y } f_{S\mapsto I}(t) = 0 \\ S & \text{si } E_i(t) = S \text{ y } f_{S\mapsto R}(t) = 0 \\ S & \text{si } E_i(t) = R \text{ y } f_{R\mapsto S}(t) = 1 \\ I & \text{si } E_i(t) = S \text{ y } f_{S\mapsto I}(t) = 1 \\ I & \text{si } E_i(t) = I \text{ y } f_{I\mapsto R}(t) = 0 \\ R & \text{si } E_i(t) = S \text{ y } f_{S\mapsto R}(t) = 1 \\ R & \text{si } E_i(t) = I \text{ y } f_{I\mapsto R}(t) = 1 \\ R & \text{si } E_i(t) = R \text{ y } f_{R\mapsto S}(t) = 0 \end{cases} \tag{7}$$

donde:

$$f_{S\mapsto I}(t) = \bigwedge_{[j]\in V_i} r_{i,j}(t) \vee \Omega_i(t), \tag{8}$$

$$f_{S\mapsto R}(t) = \begin{cases} 1, & \text{con probabilidad } \phi_i(t) \\ 0, & \text{con probabilidad } 1-\phi_i(t) \end{cases} \tag{9}$$

$$f_{R\mapsto S}(t) = \begin{cases} 1, & \text{con probabilidad } \delta_i(t) \\ 0, & \text{con probabilidad } 1-\delta_i(t) \end{cases} \tag{10}$$

$$f_{I\mapsto R}(t) = s_i(t) \vee \Gamma_i(t), \tag{11}$$

siendo:

$$\Omega_i(t) = \begin{cases} 1, & \text{con probabilidad } \beta_i(t) \\ 0, & \text{con probabilidad } 1-\beta_i(t) \end{cases} \tag{12}$$

$$\Gamma_i(t) = \begin{cases} 1, & \text{con probabilidad } \gamma_i(t) \\ 0, & \text{con probabilidad } 1-\gamma_i(t) \end{cases} \tag{13}$$

IV-B. Simulaciones

A continuación se realizarán una serie de simulaciones para poder comparar los dos modelos. En estos casos y para simplificar, no tendremos en cuenta la dinámica poblacional (aparición y desaparición de ordenadores). En ellas se considerarán $n = 500$ ordenadores en la red y se supondrán que inicialmente hay 5 ordenadores infectados ($I(0) = 5$).

En primer lugar se tiene en cuenta un escenario homogéneo (condiciones de los modelos continuos), esto es, se considerará que todos los ordenadores se encuentran conectados entre sí en todo momento (es decir, la topología asociada al autómata celular viene definida por un grafo completo), y se supondrá que todos los ordenadores poseen los mismos valores de los parámetros:

$$\beta_i(t) = \beta(t), \phi_i(t) = \phi, \delta_i(t) = \delta, \gamma_i(t) = \gamma, \quad \forall i. \tag{14}$$

Concretamente usaremos los mismos valores de los parámetros que los empleados por Feng *et al.* en [3], esto es:

$$\gamma = 0{,}2, \beta = 0{,}8, \phi = 0{,}46, \delta = 0{,}7, \tau = 10. \tag{15}$$

La simulación obtenida con el modelo de Feng *et al* se muestra en la figura 1-(a), mientras en que la figura 1-(b) se presenta la simulación obtenida con el modelo discreto. Como se puede apreciar en estas simulaciones las tendencias globales obtenidas en ambos casos son similares aunque en la conseguida a partir del modelo basado en autómatas celulares (escenario individual) se puede observar cómo es más sensible a las interconexiones entre los diferentes elementos de la red. Como se ha comentado anteriormente, el modelo discreto permite obtener también la evolución individual de

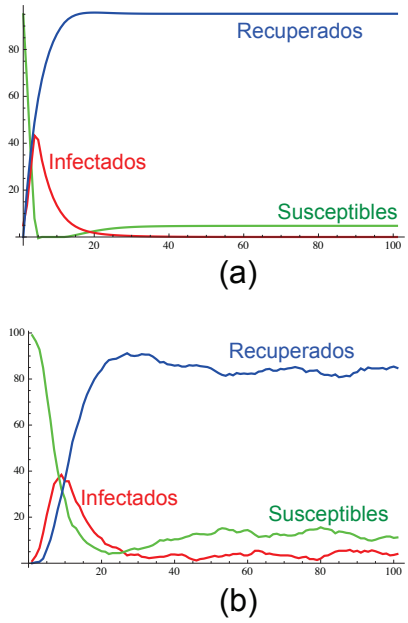


Figura 1: Evolución global de las diferentes clases en un escenario homogéneo. (a) Modelo continuo de Feng *et al.* (b) Modelo discreto propuesto en este trabajo.

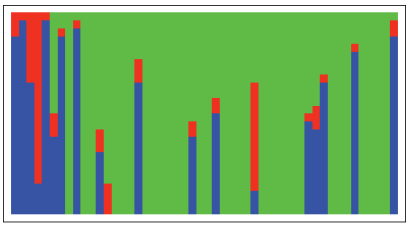


Figura 2: Evolución individual de una colección de ordenadores: cada columna representa un ordenador de manera que el estado susceptible se representa en color verde, el estado infectado en color rojo, y el estado recuperado en color azul.

cada uno de los ordenadores de la red. En la figura 2 se puede observar el diagrama de evolución de estados de una serie de ordenadores: cada columna representa la evolución de un ordenador diferente.

Por otro lado, y dentro del escenario individualizado, supondremos en primer lugar que se mantienen las conexiones según un grafo completo (todo ordenador está conectado con el resto) y que la población se divide en dos grupos atendiendo a las características y prácticas de seguridad que presentan y tienen tanto los ordenadores como sus usuarios. El tipo A estará definido por aquellos ordenadores y usuarios asociados que se preocupen por la seguridad (tengan sistemas operativos y software antivirus instalado y actualizado, tengan prácticas seguras en el uso de Internet, etc.), mientras que el tipo B lo constituirán aquellos dispositivos y usuarios con prácticas más relajadas en temas de seguridad. En la tabla II se muestra el rango de valores numéricos asignados a cada uno de los parámetros para cada uno de los tipos (estos

Tabla II: Valores de los parámetros en el escenario individualizado

Parámetro	Valores (usuarios tipo A)	Valores (usuarios tipo B)
δ_i	$0{,}25 \leq \delta_i \leq 0{,}5$	$0{,}5 \leq \delta_i \leq 0{,}75$
β_i	$0{,}25 \leq \beta_i \leq 0{,}5$	$0{,}5 \leq \beta_i \leq 0{,}75$
ϕ_i	$0{,}5 \leq \phi_i \leq 1$	$0 \leq \phi_i \leq 0{,}5$
γ_i	$0{,}5 \leq \gamma_i \leq 0{,}75$	$0{,}1 \leq \gamma_i \leq 0{,}4$
τ_i	$1 \leq \tau_i \leq 10$	$1 \leq \tau_i \leq 10$

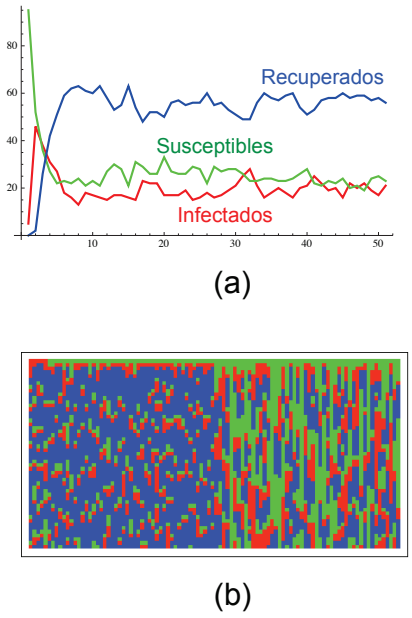


Figura 3: Evolución de las diferentes clases en un escenario individual con topología definida por un grafo completo. (a) Dinámica global (b) Dinámica individual.

valores son meramente ilustrativos). Se supondrá además que los ordenadores se reparten por igual entre los dos tipos.

En la figura 3 se puede observar la evolución tanto global (figura 3-(a)) como individual (figura 3-(b)) de los ordenadores de la red. Obsérvese que los ordenadores correspondientes al tipo A (cuya evolución viene representada por la primera mitad de columnas de la figura 3-(b)) se infectan prácticamente en la misma proporción que el resto pero se recuperan antes y permanecen en dicho estado mucho más tiempo que el resto.

Por otra parte, y dentro del escenario individualizado, supondremos a continuación que la topología de la red de ordenadores no viene definida por un grafo completo sino por el grafo que se muestra en la figura 4 (en gris se encuentran representados los ordenadores del tipo A, mientras que en negro se colorean los ordenadores correspondientes al tipo B). En este caso supondremos que los parámetros siguen lo establecido en la tabla II. En la figura 5 se ilustra la situación que presenta la red en tres instantes de tiempo: $t = 0,3$ y $t = 6$.

Obsérvese que los ordenadores del cúmulo de la izquierda (todos ellos pertenecientes al tipo A) tardan más tiempo en

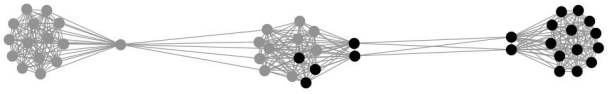


Figura 4: Grafo que define la topología de la red.

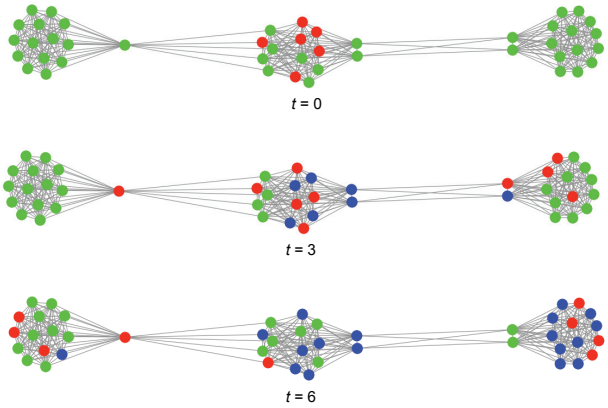


Figura 5: Evolución individual de las diferentes clases.

infectarse que el resto de los ordenadores. Asimismo, se comprueba cómo más del $50\,\%$ de los ordenadores del cúmulo de la derecha (todos ellos son del tipo B) se infectan en los 5 primeros pasos de tiempo.

V. Conclusiones

Los modelos matemáticos diseñados para simular la propagación de malware en redes de ordenadores son eminentemente de naturaleza determinista y continua, y su dinámica se basa en sistemas de ecuaciones diferenciales ordinarias.

Estos modelos, debido al paradigma en el que se fundamentan, presentan las siguientes deficiencias:

(a) Consideran que todos los ordenadores se encuentran conectados entre sí. Consecuentemente no se tienen en cuenta la conexiones locales entre los elementos de la red.

(b) No tienen en cuenta las características individuales de los ordenadores que forma la red, esto es, los parámetros de los que depende la dinámica del malware, son globales: se utilizan los mismos para todos los ordenadores.

Estos problemas se pueden solventar si basamos los modelos en otro tipo de herramientas matemáticas que permitan incorporar las características propias de cada uno de los ordenadores, a saber: tipo de sistema operativo instalado y frecuencia con la que se actualiza, tipo de software de seguridad instalado (firewall, software antivirus, etc.), concienciación del usuario en temas de seguridad, prácticas del usuario, etc.

Así se considera el uso de los autómatas celulares como posible herramienta para el diseño de dichos modelos. En este sentido se estudia el modelo basado en ecuaciones diferenciales propuesto por Feng *et al.* y se propone una alternativa al mismo basada en un autómata celular booleano. Se comprueba que las simulaciones obtenidas en el caso homogéneo (en el que se suponen las condiciones de los modelos continuos) son similares en ambos modelos cuando el número de ordenadores es elevado. Asimismo, se han realizado también simulaciones en el caso individual (cuando los valores de los parámetros varían con el ordenador) y se han mostrado tanto la evolución global como la individual (cosa que no es posible con el modelo basado en ecuaciones diferenciales). Se comprueba como el modelo basado en autómatas celulares es más sensible a las conexiones locales entre los diferentes elementos de la red; asimismo, produce resultados más ajustados a la realidad que el modelo basado en ecuaciones diferenciales cuando el número de ordenadores es pequeño.

Agradecimientos

Este trabajo ha sido subvencionado por el Ministerio de Economía y Competitividad bajo el proyecto TUERI (TIN2011-25452) y por la Consejería de Educación de la Junta de Castilla y León.

Referencias

[1] J. Amador, J.R. Artalejo, "Modeling computer virus with the BSDE approach,"*Computer Networks*, vol. 57, pp. 302-316, 2013.

[2] J. Amador, J.R. Artalejo, "Stochastic modeling of computer virus spreading with warming signals,"*Journal of the Franklin Institute*, vol. 350, pp. 1112-1138, 2013.

[3] L. Feng, X. Liao, Q. Han, H. Li, "Dynamical analysis and control strategies on malware propagation model," *Applied Mathematical Modelling*, vol. 37, pp. 8225–8236, 2013.

[4] J. Hao, J. YIN, B. Zhang, "Modeling viral agents and their dynamics with persistent turing machines and cellular automata,"*Lecture Notes in Computer Science*, vol. 4088, pp. 690-695, 2006.

[5] W.O. Kermack, A.G. McKendrick, "A Contribution to the Mathematical Theory of Epidemics,"*Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 115, pp. 700-721, 1927.

[6] S. Kondakci, "Epidemic state analysis of computers under malware attacks,"*Simulation Modelling Practice and Theory*, Vol. 16, pp. 571-584, 2008.

[7] A. Martín del Rey, "A Computer Virus Spread Model Based On Cellular Automata of Graphs,"*Lecture Notes in Computer Science*, vol. 5518, pp. 503-506, 2009.

[8] A. Martín del Rey, "A SIR e-Epidemic model for computer worms based on cellular automata,"*Lecture Notes in Artificial Intelligence*, vol. 8109, pp. 228-238, 2013.

[9] A. Martín del Rey, G. Rodríguez Sánchez, "A discrete mathematical model to simulate malware spreading,"*International Journal of Modern Physics C*, vol. 23, paper id. 1250064, 2012.

[10] M. Meisel, V. Pappas, L. Zhang, "A taxonomy of biologically inspired research in computer networking,"*Computers Networks*, vol. 54, pp. 901-916, 2010.

[11] B.K. Mishra, N. Keshri, "Mathematical model on the transmission of worms in wireless sensor network,"*Applied Mathematical Modelling*, vol. 37, pp. 4103-4111,2012.

[12] B.K. Mishra, D.K. Saini, "SEIRS epidemic model with delay for transmission of malicious objects in computer network,"*Applied Mathematics and Computation*, vol. 188, pp. 1476-1482, 2007.

[13] J. Ren, X. Yang, L.X. Yang, Y. Xu, F. Yang, "A delayed computer virus propagation model and its dynamics,"*Chaos, Solitons & Fractals*, vol. 45, pp. 74-79, 2012.

[14] M. Rice, J. Butts, R. Miller, S. Shenoi, "Applying public health strategies to the protection of cyberspace,"*International Journal of Critical Infrastructure Protection*, vol. 3, pp.118-127, 2010.

[15] O.A. Toutonji, S.M. Yoo, M. Park, "Stability analysis of VEISV propagation modeling for network worm attack,"*Applied Mathematical Modelling*, vol. 36, pp. 2751-2761, 2012.

[16] S. Wolfram, "A New Kind of Science," Champaign, IL: Wolfram Media Inc., 2002.

[17] Y. Yao, L. Guo, H. Guo, G. Yu, F.X. Gao, X.J. Tong, "Pulse quarantine strategy of internet worm propagation: Modeling and analysis,"*Computers and Electrical Engineering*,vol. 38, pp. 1047-1061, 2012.

Contramedidas en la suplantación de autoridades de certificación. Certificate pinning

Alfonso Muñoz
Telefonica Digital Identity - Privacy
Security researcher
Email: alfonso.munoz@11paths.com

Antonio Guzmán
Telefonica Digital Identity - Privacy
Security researcher
Email: antonio.guzman@11paths.com

Sergio de los Santos
Telefonica Digital Identity - Privacy
Security researcher
Email: sergio.delossantos@11paths.com

***Resumen*—La importancia de asegurar la comunicación entre personas ha crecido a medida que se ha avanzado en la sofisticación y el alcance de los mecanismos provistos para ello. Ahora, en la era digital, el alcance de estas comunicaciones es global y surge la necesidad de confiar en infraestructuras que suplan la imposibilidad de identificar a ambos extremos de la comunicación. Es la infraestructura de autoridades de certificación y la gestión correcta de certificados digitales la que ha facilitado una aproximación más eficiente para cubrir esta demanda. Existen, sin embargo, algunos aspectos de esta infraestructura o de la implementación de algunos de sus mecanismos que pueden ser aprovechados para vulnerar la seguridad que su uso debe garantizar. La presente investigación profundiza en alguno de estos aspectos y analiza la validez de las soluciones propuestas por grandes productores de software frente a escenarios realistas.**

***Palabras clave*—certificate pinning, certificado digital, identificación, confidencialidad**

I. Introducción

La presente investigación revisa algunos de los aspectos que son relativos a la seguridad de la identificación digital mediante certificados digitales. Para ello se propone un estado del arte actualizado sobre esta cuestión y se presta especial interés al concepto de *certificate pinning* y cómo éste se está utilizando en la actualidad para asegurar las comunicaciones en Internet.

Este artículo está estructurado en cinco apartados. El primer apartado refleja la estructura del artículo. El apartado segundo analiza la identificación mediante certificados digitales focalizado en tres grandes retos (criptográficos, de interfaces y de cadena de certificación), así como las contramedidas posibles a las amenazas sobre los mismos. El tercer apartado introduce el concepto de *certificate pinning*. Por último, en el apartado cuarto se reflejan nuestros estudios sobre la implementación actual del *certificate pinning* en navegadores web, concluyendo en el apartado quinto con diferentes conclusiones sobre la investigación realizada.

II. Amenazas y contramedidas definidas sobre la infraestructura para la gestión de certificados digitales

La identificación de entidades en las comunicaciones en Internet es un paso fundamental para proporcionar toda una serie de servicios apoyados en los clásicos mecanismos de seguridad, como son la confidencialidad, integridad y autenticidad, tanto de la información intercambiada como de los actores que participan en una comunicación. Hoy día, es posible establecer comunicaciones seguras en Internet de una manera escalable gracias a la criptografía de clave pública, los certificados digitales, típicamente certificados X.509v3, y las infraestructuras de clave pública. Todas estas tecnologías y algoritmos permiten a un navegador web, utilizando protocolo HTTP/TLS=HTTPS, establecer comunicaciones seguras garantizando su confidencialidad (se negocia una clave simétrica para cifrar la información), integridad y autenticidad. En la actualidad, escenarios tan importantes como el comercio electrónico o la firma electrónica no serían posibles sin estas garantías. Es tal la importancia de estas tecnologías que es vital analizar las amenazas existentes que pudieran vulnerar sus principios, así como proponer contramedidas siempre que fuera posible.

II-A. Amenazas

En la actualidad, las amenazas vienen fundamentalmente de tres vías: problemas en la implementación de los mecanismos que garantizan estos escenarios o debilidades en los algoritmos utilizados en la gestión de los certificados digitales, vulnerabilidades de la interfaz de usuario y la posibilidad de suplantación de elementos en la cadena de certificación que garantiza la autoría de una clave pública.

- **Problemas derivados de fallos criptográficos e implementación**

Toda la seguridad de los certificados digitales recae en la robustez e implementación adecuada de los algoritmos criptográficos utilizados (cifrado y hash). En los últimos años se han documentado casos graves de vulneración de comunicaciones cuando esto no se produce. Entre los más significativos destacan la investigación de Luciano Bello demostrando la implementación incorrecta de OpenSSL que permitía invertir procesos criptográficos [1], la investigación de Alexander Sotirov et al. [2] falsificando certificados digitales aprovechando ataques de colisión al algoritmo MD5 o la investigación de Moxie Marlinspike [3] que descubrió que las autoridades de certificación no validan adecuadamente el campo CN (Common Name), al firmar un certificado, y por tanto era factible utilizar una codificación especial para hacer enmascarar un dominio ilegítimo haciendo pensar al usuario que está viendo uno legítimo.

- **Vulnerabilidades de las interfaces de usuario**

Los terminales que se utilizan para comunicaciones en Internet suelen ser terminales inseguros por definición, clásicamente, el computador de sobremesa o los dispositivos móviles. Si es posible manipular su configuración o instalar malware, cualquier sistema de seguridad será anulado. Por ejemplo, en el caso del eDNI, esto ha sido puesto de manifiesto en diversas ocasiones [4]. Si se parte del supuesto que la seguridad del sistema no se ha comprometido, entonces se puede concluir que parte de las amenazas estarán basadas en que es posible engañar al usuario aprovechándose de particularidades de la interfaz de usuario (su configuración y el modo que opera el usuario con ella), es decir, típicamente el navegador web. Posiblemente el ejemplo más representativo es el intento de hacer aceptar a un usuario un certificado como válido cuando el navegador le indica que no lo es (al final es decisión del usuario aceptarlo o rechazarlo). Por desgracia, en determinados escenarios esta decisión podría ser transparente al usuario y no ser consciente de la suplantación. El investigador Moxie Marlinspike descubrió en 2009 que una configuración incorrecta del protocolo OCSP (Online Certificate Status Protocol) simplificaría ataques al protocolo SSL [5]. OCSP es un protocolo de consulta online para saber si un determinado certificado digital ha sido revocado o no. Para ello, el cliente envía la petición a la dirección de la CRL (Certificate Revocation List), que viene indicada en el propio certificado digital. Si un atacante está haciendo un ataque de hombre en el medio para utilizar uno de estos certificados digitales, entonces también puede interceptar las peticiones OCSP y utilizarlas en su provecho. En un funcionamiento normal, un servidor mediante este protocolo podría enviar una respuesta *Try Later* indicando al cliente que ahora no puede atender una petición. El atacante podría simular esta contestación, que tiene asignado el código 3, para indicar al cliente que ahora no puede atender su petición. Ante esta situación muchos clientes web aceptaban el certificado digital al no poder corroborar su validez. En la práctica muchos esfuerzos se han realizado en el pasado, ataques y herramientas, para engañar al usuario. Un ejemplo significativo es la herramienta SSLstrip, del investigador Moxie Marlinskpe [6], que intenta engañar al usuario de la siguiente forma: cuando se llama a una página web, se sustituyen todos los enlaces https por http, con la intención que la comunicación entre el cliente y el atacante sea por http y la comunicación entre atacante y servidor por https. Para engañar a usuarios menos formados se simula el *candado amarillo* cargando esta imagen en el favicon. En los últimos años se ha documentado también un especial interés en la detección de malas implementaciones por parte del interfaz del usuario, navegador web, de protocolos de seguridad. Los ataques más significativos han sido BEAST [7], CRIME [8] y BREACH [9], que se apoyaban en implementaciones inadecuadas del protocolo TLS/SSL permitiendo, bajo ciertas condiciones, descifrar una comunicación (cookies de sesión).

- **Suplantación de la cadena de certificación**

La seguridad en Internet se apoya en la confianza en autoridades intermedias que certificarán la autenticidad de un certificado digital. Esta confianza ejecutada de manera recursiva a diferentes niveles, hasta resolver la autenticidad de un certificado digital de un usuario, se conoce como cadena de certificación. En los últimos años multitud de ataques se han centrado en suplantar a autoridades o certificados pertenecientes a la cadena de certificación con un objetivo claro: un certificado falso validado por la cadena de certificación será dado como bueno y por tanto se podrán suplantar dominios válidos. Algunos sucesos significativos han sido: compromiso de la CA Comodo, compromiso CA Diginotar [10], CA TurkTrust [11], virus Flame [12] (firmado de código y suplantación de Windows update), certificado Adobe (firmado de herramientas ilegítimas) [13], etc.

II-B. Contramedidas

Las amenazas reflejadas en el apartado anterior tienen soluciones diferentes. Las dos primeras amenazas aunque no resueltas de manera global pueden ser mitigadas con buenas prácticas y sistemas robustos de actualización. Por ejemplo, en el caso de la navegación segura en Internet utilizando el protocolo HTTP/SSL existen recomendaciones que son necesarias conocer [14]. No obstante, el usuario siempre puede tomar algunas medidas adicionales, que aunque requieran cierta configuración, pueden ser de utilidad en la mitigación. Por ejemplo, el uso de extensiones en el navegador web para forzar siempre de manera automática la conexión https a un dominio si ésta existe (add-on Firefox https everywhere [15]), la configuración adecuada de los protocolos que verifican el estado de revocación de un certificado digital o una postura más activa que permita al usuario comprobar la seguridad del servidor web con el que se comunica. Para ello una herramienta de gran utilidad es la herramienta TLSSLed [16].

Por otro lado, si se centra la atención en la última amenaza destacada en el apartado anterior, "suplantación de elementos en la cadena de certificación", la bibliografía publicada en cuanto a contramedidas y la evolución de las mismas en entornos reales es, a nuestro entender, escasa. En la actualidad este tema se ha abordado de manera individual en los extremos de la comunicación. Desde el punto de vista del desarrollador de una PKI, cómo implementarla y protegerla para evitar suplantaciones [17] [18]. Desde el punto de vista del usuario instalar y configurar contramedidas para detectar posibles plagios y modificaciones en certificados digitales. Un ejemplo de lo anterior consistiría en reducir los vectores de ataque bloqueando autoridades de certificación en función de su procedencia geográfica [20]. Esto minimiza el impacto de certificados (y por tanto dominios web) firmados por organizaciones que podrían ser más sobornables o manipulables en determinados países. En el caso más extremo, algunos investigadores como Moxie Marlinspike han propuesto, con la extensión Firefox denominada Convergence [21], no sin inconvenientes, delegar la confianza de si un certificado es válido en la decisión de un número de usuarios en red,

conceptualmente en una aproximación similar al concepto de anillo de claves de confianza en GPG.

De todas las contramedidas documentadas una técnica que está comenzando a mostrar gran utilidad es garantizar de manera transparente y automatizada la cadena de certificación para evitar ataques derivados de suplantación de entidades intermedias. Esta contramedida se conoce como *certificate pinning*.

III. Certificate Pinning. Reduciendo vectores de ataque

El problema principal en la suplantación de autoridades intermedias en una cadena de certificación, necesaria para validar la autenticidad de un certificado digital asociada a un dominio, reside en que los navegadores web no tienen la capacidad, por defecto, de detectar que la cadena de confianza se ha modificado (que no comprometido), ya que el certificado/CA comprometido estará firmado por una CA válida que a su vez depende de una CA de nivel superior de la cual se confía y no ha sido comprometida.

La idea detrás del *certificate pinning* reside precisamente en poder detectar cuándo una cadena de confianza ha sido modificada. Para ello se busca asociar inequívocamente un certificado digital a un dominio concreto (se recuerda certificados presentes en una cadena de certificación). De esta forma, un dominio A, por ejemplo www.google.com, estará vinculado a un certificado/autoridad de certificación B específico. Si una autoridad de certificación B' diferente (que depende de una autoridad de certificación raíz de la que se confía) intenta emitir un certificado asociado al dominio A este hecho generará una alerta. La cadena de certificación no ha tenido por qué ser comprometida, pero sí se detecta que ha sido modificada. Esta característica hubiera permitido detectar ataques recientes derivados del compromiso de entidades intermedias, como fue el caso del compromiso de la CA de Comodo [10].

En la práctica, a día de hoy no existe consenso en cómo llevar estos principios al mundo real. Aunque todavía no existe ningún estándar se están comenzando a vislumbrar trabajos más maduros en esta dirección [22][25][26]. En febrero del 2014 se publicó un borrador de RFC [22] en el que el IETF estudia la posibilidad de que el concepto de *certificate pinning* se incluya directamente en el protocolo HTTP. Con esta futura modificación al protocolo los dominios enviarían información HTTP al navegador sobre sus certificados utilizando cabeceras HTTP:

```
Public-Key-Pins:
pin-sha1=''4n972HfV354KP560yw4uqe/baXc='';
pin-sha1=''qvTGHdzF6KLavt4PO0gs2a6pQ00='';
pin-sha256=
''LPJNul+wow4m6DsqxbninhsWHlwfp0JecwQzYpOLm\\CQ='';
max-age=10000;
includeSubDomains
```

En esta propuesta de RFC, en esencia, se propone enviar un hash por clave pública a *pinear* (pin-sha1 o pin-sha256) y el tiempo máximo (max-age) en el cual se debería confiar en esa información. En cualquier caso, y mientras se extiende su utilización, las soluciones existentes se centran en la realización de cambios, más o menos complejos, a los navegadores y protocolos utilizados para acercarse al concepto de *pinning*. Entre las propuestas analizadas son destacables:

Proyecto DANE (DNS-Based Authentication of Named Entities) [26]. El IETF normaliza en la RFC 6698 la posibilidad de vincular el protocolo TLS a dominios específicos realizando ciertas modificaciones al protocolo DNS. Lógicamente se debe confiar en la validez del firmador de ese certificado, y como ya se ha visto éste es uno de los problemas actuales de las autoridades de certificación en Internet. Una aproximación a la solución de este problema consistiría en la utilización de DNSSEC (DNS Security Extensions), pudiendo vincular claves criptográficas (certificados) a nombres de dominio DNS (en lugar de a cadenas de texto más o menos arbitrarias presentes en un certificado). En este sentido, el proyecto DANE (DNS-Based Authentication of Named Entities) proporciona la posibilidad de utilizar la infraestructura DNSSEC para almacenar y firmar claves/certificados que serán utilizados en TLS.

Proyecto TACKS (Trust Assertions for Certificate Key) [27]. Moxie Marlinspike y T. Perrie en su propuesta TACKS (internet-draft) proponen una extensión al propio protocolo TLS para permitir el registro de la cadena de certificación de los certificados digitales. La idea es que un cliente intentando conectar a un servidor protegido pudiera resolver esta asociación a nivel de conexión TLS.

IV. Analisis de implementaciones de certificate pinning. Navegadores web

Las propuestas reflejadas en el apartado anterior muestran diferentes intentos para llevar a la práctica el concepto del *certificate pinning* como contramedida para los problemas derivados de ataques a la cadena de certificación. Por desgracia, como se ha indicado anteriormente, no existe a día de hoy una solución estándar, aunque diferentes fabricantes están realizando implementaciones propietarias buscando las ventajas de esta contramedida. En este apartado se indaga en la implementación del *certificate pinning* en algunos de los navegadores más utilizados, destacando sus ventajas e inconvenientes.

Para el análisis de todos ellos se parte del siguiente escenario genérico de prueba. En el cual se considera que las comunicaciones que utilicen protocolo SSL utilizarán certificados digitales que podrán ser validados por una cadena de certificación. Típicamente una autoridad certificadora hoja y raíz, y una o más autoridades de certificación intermedias.

Este escenario genérico implementado en nuestros análisis permitirá analizar diferentes escenarios donde la cadena de certificación, sin comprometerse, se modifica de alguna forma. Escenarios clásicos son cuando o el certificado hoja cambia por algún motivo o alguna autoridad certificadora intermedia

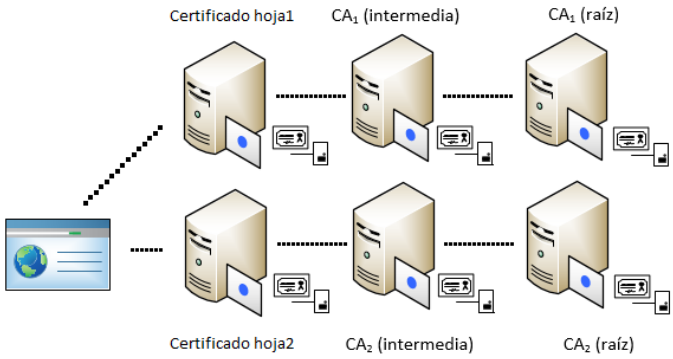


Figura 1. Escenario genérico de prueba con autoridades de certificación

es comprometida. Por ejemplo, un escenario de prueba es en el que originalmente el navegador confía en las autoridades certificadoras raíz (CA1 y CA2) y se compromete la autoridad certificadora intermedia CA2, generando un certificado para el dominio X. Dominio y certificado que fue generado originalmente por la autoridad intermedia CA1. Este escenario ha sido el más habitual en los ataques publicados en los últimos años [10].

IV-A. Microsoft Internet Explorer

Microsoft implementa una aproximación a la funcionalidad de *certificate pinning* en su herramienta de seguridad EMET [28]. El kit de herramientas de experiencia de mitigación mejorada (EMET) es una utilidad que ayuda a prevenir la explotación de vulnerabilidades de seguridad en el software.

Esta herramienta se aproxima al concepto de *certificate pinning* facilitando la unión de dominios con certificados raíz del repositorio de certificados de confianza del usuario o de la máquina. Mediante la herramienta Process Explorer [29] se puede observar cómo se inyecta la librería EMET_CE.DLL en el proceso de Internet Explorer lo que permite, mediante su configuración, analizar si el dominio al que supuestamente pertenece un certificado digital coincide con una regla de configuración definida en EMET. Si no fuera así se avisa con una pequeña ventana emergente. Por desgracia, la apuesta de Microsoft resulta ser un mecanismo poco usable para usuarios no formados.

Aunque por defecto vienen preconfigurados algunos dominios populares (Facebook, Twitter, etc.) esta solución todavía presenta una serie de inconvenientes reseñables:

- EMET realiza *pineo* exclusivamente a certificados creados por autoridades raíz registradas en el almacén de certificados del sistema operativo Windows. Cualquier compromiso de una CA intermedia/hoja que tenga una CA raíz de la cual se confía no podrá ser detectado. Por tanto, se podrían generar certificados para dominios válidos, por ejemplo www.google.com, desde una CA que no sea la creadora original del mismo (cadena de certificación modificada aunque no comprometida).
- Esta solución no está integrada directamente en el navegador web y su usabilidad es cuestionable. Requiere configuración local y manual. La solución debe instalarse y configurarse explícitamente. Esta tarea puede simplificarse mediante la herramienta EmetRules [30].
- EMET podría ser utilizado con otros navegadores pero suele estar contraindicado. Por ejemplo, en el caso del navegador Chrome, la recomendación oficial es no utilizarlo ya que afecta negativamente al rendimiento y no proporciona mayor seguridad que las contramedidas implementadas por defecto en el navegador [31].

IV-B. Google Chrome

Google aborda el problema de la confianza en un certificado digital combinando dos principios en su navegador web: HSTS y *pineo* de certificados.

Chrome implementa el estándar HSTS (Http Strict Transport Security). Se trata de una especificación que permite obligar a que, en una página, se use siempre https aunque el usuario no lo escriba en la barra del navegador. La idea fundamental es que el servidor web mediante cabeceras http fuerce al navegador web a conectarse directamente por https. Para ello, el servidor que lo desee debe enviarle una cabecera al navegador, del tipo:

```
Strict-Transport-Security:
    max-age=16070400;
    includeSubDomains
```

Esta característica aunque interesante tiene un problema. Si la primera vez que se realiza una conexión a un servidor (o cuando la información enviada expire) se realiza en una red hostil, ataque de hombre en el medio, se demuestra que la información enviada vía cabeceras http puede ser suprimida o modificada, engañando al navegador (hasta que expire la información: max-age). Precisamente por este motivo es útil utilizar una protección extra que propone Chrome y es la posibilidad de incorporar dominios bajo petición. La idea es que Chrome bajo petición (alg@chromium.com) incorpora en su código fuente [32] forzar la conexión HTTPS para los dominios reflejados. Es los que se conoce como *Preloaded HSTS sites*. Lógicamente esta solución no parece escalable a largo plazo.

En cualquier caso, aunque interesante, la protección anterior no protege frente a un escenario de ataque basado en certificados intermedios diferentes o emitidos de forma fraudulenta. La cadena de certificación se ha modificado pero no se ha roto. En este caso Chrome, complementa la medida anterior, con una aproximación al concepto de *certificate pinning*.

Antes de analizar las formas en las que Chrome puede realizar *pinneo* es importante entender cómo valida los certificados digitales con los que opera.

En la validación de certificados Chrome no comprueba el certificado completo para calcular su validez, sólo la clave

pública asociada a él (SubjectPublicKey+ SubjectPublicKeyInfo). Tradicionalmente los navegadores, y otras herramientas, calculan el hash del certificado completo como huella digital (fingerprint) para identificarlo, esto incluye el hash de todos y cada uno de los datos del certificado, no el hash de la clave pública en sí. Por ejemplo, si se cambia la fecha de caducidad o cualquier otro dato Chrome no se dará cuenta. Esto no es necesariamente negativo, en función del entorno esta flexibilidad podría evitar falsas alarmas.

Una vez resuelto cómo valida Chrome los certificados es importante responder a la siguiente pregunta ¿cómo realiza el *pinneo*? Chrome soporta *pineo* de certificados mediante dos mecanismos:

1. Servidores que soportan el borrador de RFC [22]. Mediante envío de cabeceras HTTP un servidor enviará a un navegador información de los certificados que tiene que recordar y por tanto *pinear* (en este escenario surge el mismo problema de posible ataque MitM visto con HSTS). Por ejemplo:

```
curl -i --insecure https://stalkr.net
Strict-Transport-Security: max-age=315360000
Public-Key-Pins: max-age=31536000;
pin-sha256="byhRQJ1xBQSjURWry5pt0/JXfsk3ye8ZO
```

2. *Pineo* establecido en el código fuente del navegador. En *https://src.chromium.org/svn/branches/1312/src/net/base/transport_security_state_static.h* puede comprobarse como Chrome *pinea* por defecto una serie de certificados digitales almacenando el hash de la clave pública, más exactamente el hash sha1 en base64 con el SubjectPublicKeyInfo y la propia clave pública. Por ejemplo: VeriSignClass3, Google2048, GeoTrustGlobal, etc. Al menos, *pinea* autoridades de certificación responsables de certificados digitales de servicios variados de Google.

Debe tenerse en cuenta que Chrome ignorará el *pineo* si un certificado raíz es instalado por el usuario. La razón fundamental recae en posibilitar instalar certificados cuando soluciones antivirus o proxies instalados en entornos corporativos necesitan inspeccionar el tráfico SSL [24].

Por tanto, conocido lo anterior, ¿qué certificados puede *pinear* Chrome? En principio, Chrome podría asociar un dominio a cualquier certificado en cualquier punto de la cadena de certificación. En la práctica, se demuestra que Chrome solo *pinea* el certificado de una autoridad certificadora intermedia y el de una autoridad certificadora de backup. Es posible comprobar este hecho introduciendo la siguiente url chrome://net-internals/#hsts en el navegador Chrome y realizando consultas a un dominio deseado. Por ejemplo, para www.google.com se observa: *domain:google.com pubkey_hashes: sha1/vq7OyjSnqOco9nyMCDGdy77eijM=, sha1/Q9rWMO5T+KmAym79hfRqo3mQ4Oo=*.

Este análisis permite inferir las siguientes conclusiones respecto a este navegador:

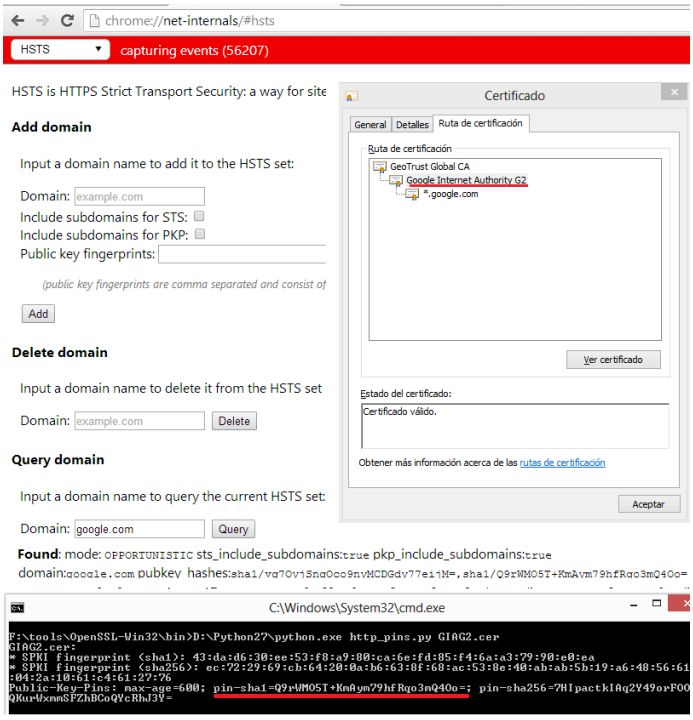


Figura 2. Detección de certificado pineado por Google

- **¿Detecta Chrome autoridades de certificación suplantadas?**

El objetivo de Chrome es intentar detectar cuándo un dominio válido, emitido originalmente por una autoridad certificadora intermedia (la cual se pinea), es emitido por otra autoridad certificadora diferente pero en la que también se confía porque se confía en una autoridad superior (típicamente raíz). Es posible detectar este escenario de cadena de certificación modificada pero no por ello comprometida, lo que supone un escenario real, y de gran utilidad como resultado de los recientes ataques documentados [10].

- **¿Qué cambios no detecta Chrome?**

Chrome, por defecto, no detecta cambios en ningún elemento de la cadena diferente a la autoridad de certificación intermedia que *pinea*. Por tanto, cualquier ataque que modifique la cadena a nivel de certificado hoja, raíz o cualquier otro elemento intermedio no será detectado. Para minimizar este efecto los servidores deberían implementar los principios del borrador de RFC [22] y enviar información de *pineo* al navegador.

IV-C. Firefox

Firefox no implementa ningún mecanismo intrínseco de *pineo* de certificados. Lo más parecido a esta aproximación es el uso del add-on *Certificate Patrol* [19] que monitoriza los cambios en certificados digitales de servidores a los que un usuario se conecta habitualmente. De todos los navegadores analizados es la única herramienta que alerta de cualquier

modificación en la cadena de certificación. Aunque esto puede resultar muy interesante tiene varios inconvenientes claros:

a) En función de la complejidad de la organización que proporcione acceso web a un servicio es común que los certificados hoja cambien por múltiples cuestiones, por ejemplo, varios certificados para balanceo de carga, etc. No parece muy conveniente el *pineo* de certificados hoja. Alertar el cambio puede saturar al usuario con alertas, muchas de las cuales no sabrá discernir como ataque.

b) La herramienta recuerda el primer certificado visto. Si se conecta por primera vez a una servidor web desde una red hostil, hombre en el medio que inyecta tráfico, se podría recordar un certificado inválido.

V. Conclusiones

El presente artículo establece un estudio del estado arte centrado en la problemática de ataques a la identificación basada en certificados digitales. El estudio se centra en el análisis y experimentación de las protecciones actuales frente a la modificación, que no compromiso, de las cadenas de certificación que certificarán o no la validez de un certificado digital.

Se ha analizado las herramientas más comunes de comunicación web en Internet, navegadores web, y se demuestra que ninguna de ellas está exenta de problemas ni permite solucionar de manera completa el problema del compromiso de cadenas de certificación. La propuesta más realista es la implementada por el navegador web Chrome si se tiene en mente una serie de limitaciones. Se ha comprobado que muchas webs populares a nivel mundial su cadena de certificación está compuesta solo por 3 niveles: certificado hoja, autoridad intermedia y autoridad raíz. Si se presupone que en general *pinear* un certificado hoja es mala idea porque puede cambiar en entornos reales (varios certificados para balanceo de carga, etc.) y una autoridad raíz debería ser muy difícil de comprometer, *pinear* solo la autoridad intermedia puede ser una solución escalable y transparente al usuario. Al final, se define un capa más de protección que, aunque no perfecta, proporciona mayores garantías en el uso de los servicios más famosos.

Mientras diferentes organismos de normalización, entre ellos el IETF [22][23], y fabricantes intentan diseñar la mejor contramedida basada en el concepto de *certificate pinning*, a día de hoy *pinear* una cadena de certificación completa no es posible en escenarios reales (cambiantes) y si no se *pinea* toda la cadena, como se ha demostrado, quedan escenarios de ataque que no se pueden detectar.

Referencias

[1] L. Bello, M. Bertacchini, “Predictable PRNG In The Vulnerable Debian OpenSSL Package. The What And The How,” *25th Chaos Communication Congress*, Berlin, Germany. December 27-30, 2008 http://events.ccc.de/congress/2008/Fahrplan/attachments/1245_openssl-debian-broken-PRNG

[2] J. Appelbaum, A. Lenstra, D. Molnar, D. Arcne, D. Weger, ‘MD5 considered harmful today Creating a rogue CA certificate‘,” *http://www.win.tue.nl/hashclash/rogue-ca/*, December 30, 2008.

[3] M. Marlinspike, ‘New Tricks For Defeating SSL In Practice‘,” *BlackHat-DC09*, https://www.blackhat.com/presentations/bh-dc-09/Marlinspike/BlackHat-DC-09-Marlinspike-Defeating-SSL.pdf

[4] G.Gonzalez, ‘Man-in-remote: PKCS11 for fun and non-profit‘,” *RootedCon 2011*, 3-5 Marzo 2011 Madrid. http://www.slideshare.net/rootedcon/gabriel-gonzalez-maninremote-pkcs11-for-fun-and-nonprofit-rootedcon-2011

[5] M. Marlinspike, ’Defeating OCSP with the character ’3”,” ,Julio 2009. http://www.thoughtcrime.org/papers/ocsp-attack.pdf

[6] M. Marlinspike, ’more tricks for defeating ssl in practice’,” *Defcon 17*, 2009. https://www.defcon.org/html/links/dc-archives/dc-17-archive.html

[7] J.Rizzo, T. Duong ‘BEAST: Surprising crypto attack against https‘,” *Ekoparty Security Conference 9º edición. 2011*

[8] J.Rizzo, T. Duong, ‘The Crime Attack‘,” *Ekoparty Security Conference 10º edición. 2012*

[9] A.Prado, N. Harris, Y. Gluck, ‘BREACH: Browser Reconnaissance & Exfiltration via Adaptive Compression of Hypertext‘,” http://breachattack.com/#howitworks

[10] R. Mandalia, ‘Security Breach in CA Networks -Comodo, DigiNotar, GlobalSign‘,” , April 2012, http://blog.isc2.org/isc2_blog/2012/04/test.html

[11] C. Wisniewski, ‘Turkish Certificate Authority screwup leads to attempted Google impersonation‘,” , April 2013, http://nakedsecurity.sophos.com/2013/01/04/turkish-certificate-authority-screwup-leads-to-attempted-google-impersonation/

[12] S. de los Santos, ‘TheFlame, el sueño de todo creador de malware‘,” , Junio 2012. http://unaaldia.hispasec.com/2012/06/theflame-el-sueno-de-todo-creador-de.html

[13] Security @adobe, ‘Inappropriate Use of Adobe Code Signing Certificate‘,” , Sept 2012. http://blogs.adobe.com/security/2012/09/inappropriate-use-of-adobe-code-signing-certificate.html

[14] C. Meyer, J. Schwenk, ‘Lessons Learned From Previous SSL/TLS Attacks. A Brief Chronology Of Attacks And Weaknesses‘,” https://eprint.iacr.org/2013/049.pdf

[15] EFF, ‘Https everywhere‘,” https://www.eff.org/https-everywhere

[16] R. Siles, ‘TLSLED‘,” , Feb 2013. http://www.taddong.com/en/lab.html#TLSSLED

[17] ETSI, ‘Policy requirements for certification authorities issuing qualified certificates (ETSI TS 101 456)‘,” , http://www.etsi.org/deliver/etsi_ts/101400_101499/101456/01.01.01_60/ts_101456v010101p.pdf

[18] Webtrust, ‘WebTrust Program for Certification Authorities‘,” , http://www.webtrust.org/homepage-documents/item27839.aspx

[19] C. Loesch, ‘Certificate Patrol‘,” , https://addons.mozilla.org/es/firefox/addon/certificate-patrol/

[20] Y. Jesus, ‘SSLcop tool‘,” , http://www.security-projects.com/?SSLCop

[21] M. Marlinspike, ‘Convergence‘,” http://convergence.io/index.html,

[22] C. Evans, C. Palmer, R. Sleevi, ‘Public Key Pinning Extension for HTTP‘,” https://tools.ietf.org/html/draft-ietf-websec-key-pinning

[23] B. Laurie, A. Langley, E. Kasper, ‘Certificate Transparency‘,” http://tools.ietf.org/html/draft-laurie-pki-sunlight-00

[24] ImperialViolet, ‘Public Key pinning‘,” https://www.imperialviolet.org/2011/05/04/pinning.html

[25] J. Walton, J. Steven, J. Manico, K. Wall, ‘Certificate and Public Key Pinning‘,” https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning,

[26] P. Hoffman, ‘The DNS-Based Authentication of Named Entities (DANE). Transport Layer Security (TLS) Protocol: TLSA‘,” https://www.rfc-editor.org/rfc/rfc6698.txt,

[27] M. Marlinspike, T. Perrin, ‘Tacks‘,” http://tack.io/-http://tack.io/draft.html,

[28] Microsoft, ‘Kit de herramientas de Experiencia de mitigación mejorada‘,” http://technet.microsoft.com/es-es/security/jj653751,

[29] M. Russinovich, ‘ProcessExplorer‘,” http://technet.microsoft.com/es-es/sysinternals/bb896653.aspx,

[30] S. de los santos, ‘EMET Rules‘,” https://www.elevenpaths.com/labs-tools-emetrules.html,

[31] Chromium projects, ‘Chromium and EMET‘,” http://dev.chromium.org/Home/chromium-security/chromium-and-emet

[32] Chrome, ‘Transport_security_state_static‘,” https://src.chromium.org/svn/branches/1312/src/net/base/transport_security_state_static.h,

Simulaciones Software para el Estudio de Amenazas contra Sistemas SCADA

Joaquin Garcia-Alfaro
Telecom SudParis
CNRS Samovar
UMR 5157, Evry, France
Email: joaquin.garcia-alfaro@acm.org

Cristina Romero-Tris
Universitat Rovira i Virgili
Avd Països Catalans, 26
43007 Tarragona
Email: cristina.romero@urv.cat

Jose Rubio-Hernan
Telecom SudParis
CNRS Samovar
UMR 5157, Evry, France
Email: jose.rubio_hernan@telecom-sudparis.eu

***Resumen*—El objetivo de las tecnologías SCADA (acrónimo de *Supervisory Control And Data Acquisition*), es proporcionar control remoto para la supervisión de infraestructuras críticas. Ataques contra tales sistemas suponen un riesgo importante. Nuestro interés en la temática es poder investigar mejoras en la seguridad de los sistemas SCADA, usando abstracciones a nivel de software, herramientas de simulación, dispositivos físicos y trazas de datos a partir de sistemas reales. Este artículo presenta, de manera general, algunas construcciones básicas de lo que son las tecnologías SCADA y sus componentes. Introduce, también, características generales de algunos simuladores open source disponibles. Por último, detalla limitaciones y mejoras potenciales, orientadas a completar el estudio de técnicas de detección de anomalías a nivel de señales físicas entre los componentes de sistemas SCADA.**

***Palabras clave*—Seguridad TIC, Detección de Intrusiones, Sistemas Críticos, Simulación por Ordenador, Sistemas SCADA.**

I. Introducción

SCADA es el acrónimo de Supervisory Control And Data Acquisition (Supervisión, Control y Adquisición de Datos). Los sistemas SCADA son grandes infraestructuras utilizadas para recoger y almacenar datos a distancia y en tiempo real. Estos sistemas se usan normalmente en la industria y en arquitecturas críticas, controlando procesos químicos, físicos o de transporte. Algunos ejemplos de sistemas SCADA son el suministro de agua, la generación y distribución de energía eléctrica o de gas.

Debido a su naturaleza crítica, una vulnerabilidad en la seguridad de un sistema SCADA podría tener graves consecuencias si fuera detectada por un atacante. Por esta razón, es necesario analizar los posibles ataques y estudiar las contramedidas existentes de cualquier sistema SCADA. El problema es que estos análisis no pueden realizarse sobre sistemas reales ya que el coste de reproducir los componentes SCADA es demasiado elevado, y no se puede asumir el riesgo de realizar experimentos en sistemas reales en funcionamiento. Por consiguiente, es necesario utilizar modelos teóricos y herramientas que permitan simular sistemas SCADA, posibles ataques, y contramedidas. Nuestra propuesta pasa por proponer una virtualización de sistemas SCADA para poder reproducir ataques cibernético en un entorno académico.

En este artículo, revisamos elementos tradicionales de una arquitectura SCADA, proponemos una arquitectura de estudio concreta y revisamos una solución existente para poder simular por ordenador, los distintos elementos de la arquitectura de estudio propuesta. A continuación, proponemos una extensión para poder investigar técnicas de detección de anomalías entre los componentes de las capas inferiores de la arquitectura de estudio. A ese nivel, la mayor parte de técnicas de detección requieren un tratamiento a nivel de las señales intercambiadas por dispositivos tales como sensores y actuadores. La mayor parte de las funciones en la solución de simulación estudiada se limitan a simular ataques contra dispositivos de capas superiores, tales como terminales remotos e interfaces intermedias. Nuestra extensión permite poder integrar funcionalidad adicional, a partir de otras plataformas de simulación, mediante el uso de librerías dinámicas compartidas. Como resultado final, esperamos poder poner en práctica técnicas de co-simulación, sin importar la naturaleza de los dispositivos evaluados (tanto reales como virtuales).

Organización del artículo: Las Secciones II y III definen conceptos básicos asociados con tecnologías SCADA de uso general. Las Secciones IV y V presentan una arquitectura SCADA de ejemplo, y nuestra metodología propuesta para evaluar amenazas y contramedidas. La Sección VI finaliza con las conclusiones del artículo.

II. Arquitectura de un Sistema SCADA

Presentamos en esta sección los elementos de una arquitectura SCADA típica. La bibliografía utilizada se basa en [1], [2], [3]. Asumimos que una arquitectura SCADA se compone principalmente de los siguientes elementos (representados, a modo de ejemplo, en la figura 1:

- Interfaces de usuario-máquina (en inglés, Human Machine Interfaces -HMIs-)
- Unidades de estación maestra (en inglés, Master Terminal Units -MTUs-)
- Unidades de estación remota (en inglés, Remote Terminal Units -RTUs-)

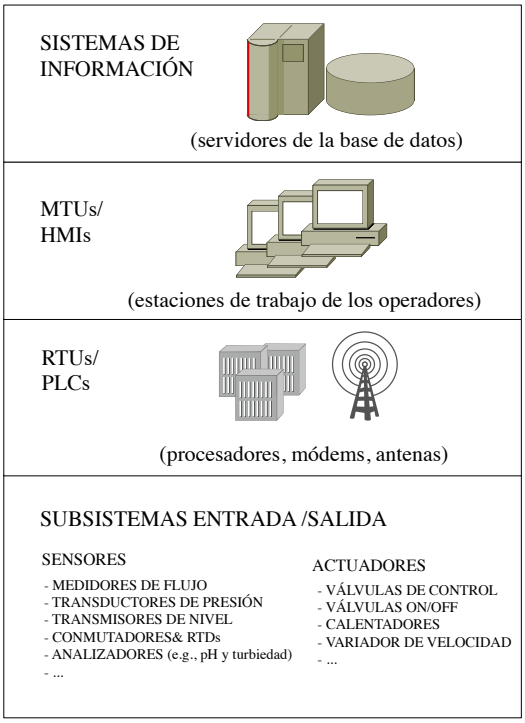


Figura 1: Elementos de un scenario SCADA de ejemplo

- Controladores lógicos programables (en inglés, Programmable Logic Controllers -PLCs-)
- Sensores y actuadores

II-A. MTUs y HMIs

Los MTUs de un sistema SCADA se localizan en el centro de control de la organización. Sirven para dar acceso a la gestión de las comunicaciones, la recogida de datos (generada por los RTUs), el almacenamiento de datos, y el control de sensores y actuadores conectados a los RTUs. La interfaz para los administradores del sistema es proporcionada por los HMIs.

II-B. RTUs

Los RTUs son unidades aisladas de adquisición de datos y control. Normalmente se trata de dispositivos basados en un microprocesador que controla y supervisa los componentes industriales de manera remota. Tienen dos tipos de funciones: (1) controlar y recoger datos de los equipos de proceso remotos, y (2) enviar los datos recogidos a una estación maestra de supervisión. Los RTUs modernos también pueden comunicarse entre ellos (ya sea con cable o de forma inalámbrica)

II-C. PLCs

Los PLCs son pequeñas máquinas de cómputo con un microprocesador. Las principales diferencias con respecto a los RTUs son el tamaño y la capacidad. Un RTU tiene un mayor número de entradas y salidas que un PLC, y mayor poder de proceso (e.g., para post-procesar los datos recogidos antes de generar las alertas para el MTU a través del HMI). Por su parte, los PLCs son frecuentemente representados como sensores con capacidad de comunicación. Los PLCs tienen dos ventajas principales respecto a los RTUs comercializados: (1) son dispositivos de uso general, permitiendo una gran variedad de funciones, y (2) son físicamente compactos, i.e., requieren menos espacio que otras alternativas.

II-D. Sensores y Actuadores

Los sensores son dispositivos de captación de medidas relacionadas con fenómenos físicos, respondiendo a algún estímulo físico. Este estímulo se transforma en una señal eléctrica, que a su vez se transforma y se almacena como datos. Los sensores pueden considerarse como el punto de entrada de un sistema SCADA. Sus datos se envían a capas superiores a través de RTUs y/o PLCs. Los actuadores son dispositivos de control, encargados de gestionar dispositivos externos. Los actuadores pueden considerarse el punto de salida de un sistema SCADA, recibiendo órdenes de RTUs y/o PLCs.

III. Protocolos de Comunicación SCADA

Los sistemas SCADA pueden usar una gran variedad de protocolos y de patrones de comunicación. A continuación se muestra un resumen de protocolos de ejemplo, respecto a los elementos descritos anteriormente.

III-A. HMI/MTUs ↔ RTUs/PLCs

La comunicación entre el centro de control (compuesto por servidores MTU/HMI) y los dispositivos remotos puede ser o no guiada. Las comunicaciones guiadas se realizan por canales eléctricos, redes de teléfono públicas (e.g., un módem de acceso telefónico o una línea alquilada) y las WANs de la organización. Las comunicaciones no guiadas se realizan por canales radio, satélite y redes inalámbricas (e.g., WPAN, WLAN, WMAM, and WWAN).

Se considera que los protocolos que se usan entre el centro de control y los dispositvos remotos son protocolos tradicionales (e.g., protocolos basados en TCP/IP), a través de redes estándares cableadas o inalámbricas (e.g., GPRS, UMTs, LTE) También es posible el uso de VPNs y de circuitos dedicados. En el centro de control, se pueden utilizar también protocolos dedicados de tipo OPC (siglas de Object Linking and Embedding (OLE) for Process Control).

III-B. RTU/PLC ↔ Sensores/Actuadores

La comunicación puede ser guiada (por cable) o no guiada (e.g., inalámbrica, basada en tecnologías como por ejemplo wifi, bluetooth, y zigbee). Protocolos de ejemplo a este nivel son Modbus (e.g., Modbus RTU y Modbus ASCII), PROFINET, DNP3 (DNP3/AGA 1.2 cifrado), EtherCAt, Fieldbus, protocolos OPC (OPC AppID y OPC UA). Remitimos al lector a consultar [1], [2], [3] para más información sobre dichos protocolos.

IV. Escenario SCADA de Ejemplo

Consideramos la arquitectura SCADA siguiente: distribución de energía, distribución de agua, y tratamiento de residuos. La figura 2 muestra una abstracción del sistema considerado.

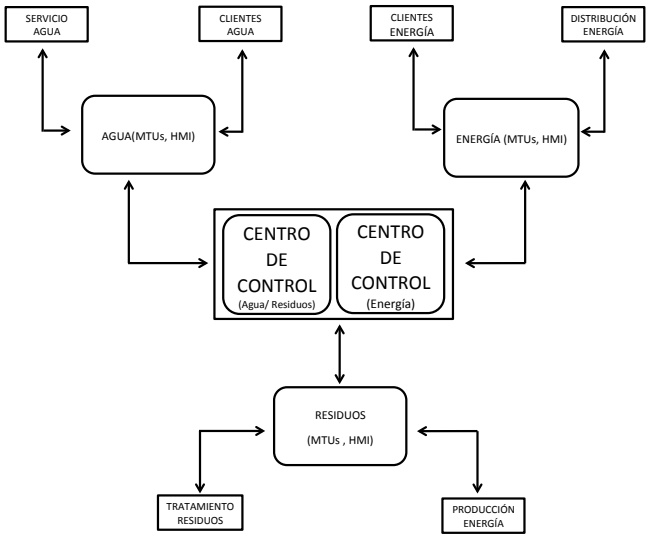


Figura 2: Escenario de ejemplo

IV-0a. Distribución de Energía: El sistema está dividido en tres capas: (a) alta tensión, (b) media tensión, y (c) baja tensión. El centro de control supervisa y gestiona tan sólo las capas (a) y (b). La capa (c) es gestionada a partir de sistemas tradicionales de tipo *hot line*.

La infraestructura asociada al centro de control cuenta con un sistema WAM (siglas en inglés de wide area measurement) que permite recoger, intercambiar y procesar los datos. El sistema WAM se complementa con una infraestructura basada en GPS para localizar los puntos finales (extremos) del sistema y un sistema operacional para gestionar los cortes de energía y la resolución de incidentes en los puntos finales.

A parte de bases de datos, asumimos aquí también elementos tipo MTUs centrales, para la gestión de áreas de alta y media tensión; puntos de suministro (e.g., energía hidráulica, solar, eólica); sub-estaciones; y puntos de transformación (eg., alta-a-media y media-a-baja) en RTUS, sensores de voltaje y corriente, y actuadores.

IV-0b. Distribución de Agua y Tratamiento de Residuos: El sistema está compuesto por un MTU central y varias sub-estaciones MTU. Cada sub-estación gestiona varios RTUs directamente conectados a sensores y actuadores. Los componentes remotes se dividen en dos capas: procesamiento (agua o residuos) y producción de energía. Los sensores (puntos de entrada del sistema) están conectados a la primera capa, proporcionando medidas de presión, temperatura, flujo y posición. Los actuadores, de carácter servomotor, están conectados al sistema de distribución de energía.

IV-0c. Protocolos de Comunicación: La comunicación (por cable o inalámbrica) desde los MTUs hasta los centros de control, así como desde los RTUs a los MTUs, utiliza VPNs a través de redes públicas y privadas (i.e., redes conmutadas públicas o líneas alquiladas para propósitos de supervisión). Asumimos que los protocolos se basan en TCP/IP. También se asume que la comunicación entre RTUs, PLCs, sensores y actuadores se realiza a través de enlaces inalámbricos o físicos. Los protocolos usados se basan, en general, en Modbus (por ejemplo, Modbus RTU o ASCII por comunicación serie o sobre TCP/IP) y DNP3 (por ejemplo, DNP3 AGA 1.2 cifrado). Remitimos al lector a consultar [1], [2], [3] para más información sobre dichos protocolos.

IV-A. Posibles Ataques a la Seguridad del Sistema

Suponemos que los objetivos del atacante son poner en riesgo la integridad y la disponibilidad del sistema dscrito. Los ataques más simples pueden ser basados en *eavesdropping*, *replay* (o ataque de reinyección), e *impersonation* (o ataque de suplantación de identidad). Ataques más complejos pueden ser iniciados como spam, phishing, e inyección de datos en puertos tipo USB. No consideramos ataques a gran escala (e.g., ataques similares a stuxnet, bien preparados, y con el apoyo técnico y financiero de grandes organizaciones). Como acciones del adversario, asumimos intercepción y modificación de paquetes, control de tráfico, inyección de comandos falsos, etc. Algunos informaciones adicionales son listadas a continuación.

- Posibles ataques a HMI/MTUs: problemas de seguridad de las tecnologías de la información y la comunicación tradicionales.
- Posibles ataques a PLCs: ataques lógicos (e.g., reescribir áreas de la memoria del PLC) y ataques físicos (e.g., apagar dispositivos de entrada/salida de manera remota).
- Posibles ataques a los puntos finales: amenazas a las comunicaciones inalámbicas, incluyendo servidores de agujero negro (blackholes), gusanos (wormholes), clonación (cloning) y suplantación de identidad (impersonation).
- Puntos de entrada: infiltración no detectada a través de dispositivos infectados (e.g., memorias USB), equipos corporativos usados de forma equivocada, protocolos vulnerables en las capas más bajas, etc.
- Problemas de seguridad en protocolos Modbus y DNP3: spoofing de mensajes en modo broadcast, ataques de replay en las respuestas a la base, control directo de esclavos, escáner de red, reconocimiento pasivo, retardo de la respuesta e intrusión.

Por último, consideramos como contramedida principal la reconfiguración del sistema (e.g., deshabilitar servicios, cortar conexiones, redirigir conexiones, bloquear aplicaciones, bajar la prioridad a los mensajes, etc.).

V. Simulación del Escenario Mediante SCADASim

Nuestra propuesta se basa en la simulación del escenario presentado en la sección IV mediante SCADASim [8], una librería para la co-simulación de entornos SCADA. A su vez,

SCADASim se basa en la plataforma de creación de simulaciones OMNeT++ [6] (disponible en http://www.omnetpp.org/). A continuación, presentamos de manera general OMNeT++ y SCADASim.

OMNeT++ es una plataforma *open source* para la creación de simuladores de eventos discretos. OMNeT++ es ampliamente utilizado a nivel académico para la simulación de redes y nuevos protocolos. La simulaciones OMNeT++ se desarrollan principalmente a partir de dos tipos de lenguaje. En primer lugar, la lógica de la simulación se desarrolla en lenguaje C++. En segundo lugar, la descripción de topologías mediante un lenguaje propio de OMNeT++ denominado NED (NEtwork Description). NED es utilizado para ensamblar componentes individuales en nuevos componentes y modelos.

Adicionalmente, OMNeT++ dispone de un gran número de librerias externas para modelar un gran número de tecnologías y protocolos, así como la creación de componentes basados en multiprocesadores y sistemas paralelos o distribuidos. Véase, por ejemplo, las librerías INET (http://inet.omnetpp.org/) para la simulación de protocolos basados en UDP, TCP, SCTP, IP, IPv6, Ethernet, PPP, 802.11, MPLS, OSPF, y muchos otros; VEINS (http://veins.car2x.org/) para la simulación de redes vehiculares; CASTALIA (http://castalia.research.nicta.com.au), para la simulación de redes inalambricas de sensores; etc. OMNeT++ permite cubrir el vacío entre herramientas orientadas a la investigación académica (tipo NS-2 y NS-3) con la potencia y facilidad de uso de herramientas comerciales de alto coste como OPNeT de Riverbed Technology (http://www.opnet.com/).

SCADASim es un proyecto *open source* disponible en http://github.com/caxqueiroz/scadasim. SCADASim permite co-simulación. Es decir, permite la coexistencia entre dispositivos simulados y dispositivos reales. SCADASim también ofrece la posibilidad de simular la ejecución de ataques contra los dispositivos de la simulación (virtuales o reales). Para ello, SCADASim tiene implementado un módulo llamado SSProxy, que actúa de enlance entre los componentes reales y los simulados. El objetivo de este módulo es recibir peticiones de una IP externa y transmitirlas a los componentes internos de la simulación. Del mismo modo, este módulo también enviaría mensajes al exterior generados por componentes internos. Para más información sobre SCADASim, remitimos al lector a las siguientes publicaciones indicadas en [8], [9].

La principal limitación actual de SCADASim es el tratamiento de la capa física de un sistema SCADA simulado por software. Por ejemplo, la funcionalidad existente para la incorporación de procesadores digitales para el tratamiento de señales es muy limitado. Esta limitación supone que el estudio de amenazas hacia las capas más bajas de la arquitectura SCADA (en especial en lo que respecta a las comunicaciones entre sensores, actuadores y PLCs), es extremadamente limitada. De hecho, esta limitación dificulta la incorporación de técnicas relevantes basada en detección de anomalías a nivel de tratamiento de señales. A modo de ejemplo, resumimos en la siguiente sección dos trabajos dentro de dicha categoria, que consideramos relevantes para el estudio de seguridad propuesto en la sección IV.

V-A. Detectores de Mo et al.

Para la detección de determinados ataques contra sistemas SCADA a nivel de PLCs, sensores y actuadores, se puede añadir a las medidas clásicas de control del flujo de datos, otro control que consiste en la autenticación de la señal que llega al sistema. Para el control y la verificación de la señal que llega al sistema, además del control y detección mediante estimación de fallos utilizado en los sistemas clásicos que permite enmarcar la señal de llegada dentro de unos patrones y verificar que la señal se sitúa dentro de los margenes (método que se puede utilizar para detectar, por ejemplo, un ataque de denegación de servicio), existen técnicas basadas en la incorporación de ruido aleatorio. Dicho ruido es introducido en la señal, para su posterior autenticación. Esta incorporación permitirá más adelante verificar que son señales válidas. En la figura 3 mostramos un detector de ejemplo propuesto por Mo *et al.* en [13], [14]. Este detector permite tratar ataques

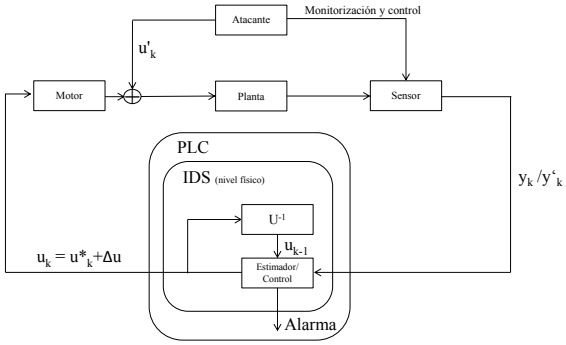


Figura 3: Sistema de detección

En la figura 3, y_k es la respuesta real del sensor, e y'_k es la respuesta forzada por el atacante al realizar un ataque de *replay*. u_k es la salida del PLC hacia el motor, y u'_k es la señal que el atacante envía la PLC como respuesta del motor. La salida del PLC puede representarse como $u_k = u^*_k + \Delta u_k$ siendo u^*_k la respuesta de control del IDS (sistema de deteción de intrusos) de nivel físico del PLC, y Δu_k el ruido aleatorio añadido en la mejora.

Una mejora en el uso de este sistema la podemos ver en [15], el sistema esta representado en la figura 4, donde se utilizan una serie de *juegos estocásticos* que permiten crear una politica de combinación entre un coste óptimo pero un sistema inseguro y un coste alto y un sistema seguro, permitiendo así mejorar el rendimiento del sistema.

V-B. Integración de los Detectores de Mo et al. en el Escenario de Ejemplo mediante SCADASim y MATLAB

En este apartado, mostramos una solución (en curso) para tratar las limitaciones de SCADASim reportadas en los apartados anteriores de esta sección. Nuestra propuesta se

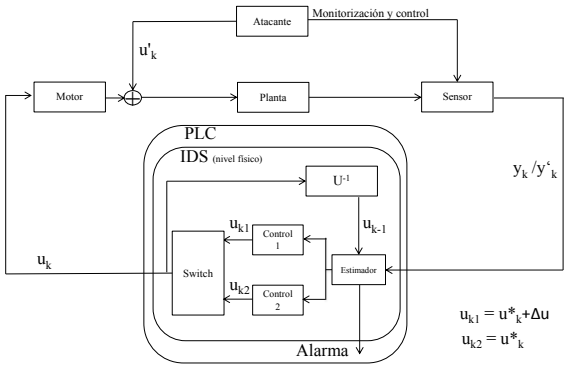


Figura 4: Detector de Mo *et al.* optimizado

basa en la incorporación de nueva funcionalidad para el procesamiento de señales digitales mediante la incorporación de librerías dinámicas mediante compilación compartida de código OMNeT++ y código MATLAB [12]. La figura 5 representa la arquitectura de nuestra propuesta. Los módulos OMNeT++ están representados en forma de cajas en color blanco. Los modulos SCADASim en cajas de color gris claro. Los módulos MATLAB en cajas de color gris oscuro. Nótese que las aplicaciones y las capas de enlace y transporte se han implementado como componentes SCADASim y OMNeT++. Las capas físicas se han implementado como librerías dinámicas MATLAB. Estás librerías son llamadas en tiempo de ejecución por las distintas instancias de OMNeT++. La capa de aplicación simplemente crea y recibe mensajes. La capa de transporte y la capa de enlace se limitan a tratar y verificar los mensajes. Por último, las funciones MATLAB son utilizadas para modular y demodular los mensajes mediante PSK (desplazamiento de fase, del inglés *Phase Shift Keying*), así como para implementar las propuestas de detección de anomalías especificadas en los trabajos de Mo et al. [13], [14].

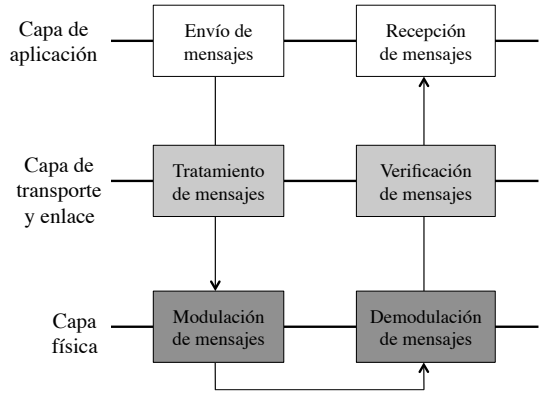


Figura 5: Arquitectura de nuestra propuesta de virtualización. Los módulos OMNeT++ están representados en forma de cajas en color blanco. Los modulos SCADASim en cajas de color gris claro. Los módulos MATLAB en cajas de color gris oscuro.

VI. Conclusiones

El objetivo de las tecnologías SCADA (Supervisory Control and Data Acquisition) es proporcionar control remoto que permita supervisar y monitorizar infraestructuras críticas e industriales, como la distribución de energía y agua. Ataques a este tipo de sistema tendrían consecuencias muy graves, y por ello es necesario herramientas que permitan detectarlos y analizar las contramedidas necesarias. En este artículo, hemos revisado una propuesta para simular ataques realizados contra sistemas SCADA, llamada SCADASim [8], [9]. A partir de la descripción de una infraestructura SCADA de ejemplo para controlar suministro de energía, hemos propuesto posibles ataques contra su seguridad, y simulados mediante SCADASim. Hemos identificado también algunas limitaciones en la versión actual de SCADASim, que impide la simulación completa de protocolos previstos en la arquitectura de ejemplo, así como la incorporación de técnicas de detección de anomalías mediante tratamiento de señales. Por ello, hemos reportado una extensión en curso sobre SCADASim, mediante la incorporación de compilación de librerías dinámicas compartidas con otras plataformas de simulación más adecuadas para el tratamiento de señales, como es el caso de MATLAB [12]. Nuestro trabajo futuro supone completar la extensión reportada en este artículo. En paralelo, proponemos trabajar otras extensiones sobre SCADASim que faciliten mayor facilidad para realizar estudios de co-simulación con componentes reales.

Referencias

[1] S. Sosik, “SCADA Systems in Wastewater Treatment,” *Process-Logic*, Whitepaper, 2007.

[2] K. Stouffer, J. Falco, K. Kent, “Guide to Industrial Control Systems (ICS) Security Recommendations of the National Institute of Standards and Technology,” *NIST Special Publication*, vol. 800, 2008.

[3] M. Shahraeini, M. H. Javidi, “SCADA Systems in Wastewater Treatment,” *Wide Area Measurement Systems. Advanced Topics in Measurements*, Chapter 15, pages 303-322, 2012.

[4] P. Manadhata, “An Attack Surface Metric,” *PhD thesis, School of Computer Science, Carnegie Mellon University*, 2008.

[5] G. Gonzalez-Ganadillo, “Optimization of Cost-based Threat Response for Security Information and Event Management Systems,” *PhD thesis, Paris VI University*, 2013.

[6] OMNeT++ Network Simulation Framework. Available On-Line. http://www.omnetpp.org/

[7] The NS-3 discrete-event network simulator. Available On-Line. http://www.nsnam.org/

[8] C. Queiroz, A. Mahmood, Z. Tari, “SCADASim – A Framework for Building SCADA Simulations,” *IEEE TRANSACTIONS ON SMART GRID*, 2(4):589–597, 2011.

[9] C. Queiroz, “A Holistic Approach for Measuring the Survivability of SCADA Systems,” PhD Thesis, College of Science, Engineering and Health, RMIT University, August 2012.

[10] M. Roesch, “Snort: Lightweight Intrusion Detection for Networks,”*13th USENIX Conference on Large Installation Systems Administration Conference (LISA-99)*, pages 229–238, 1999.

[11] Quickdraw SCADA IDS Signatures. Available On-Line. http://www.digitalbond.com/tools/quickdraw

[12] Guide, “MATLAB User’s”. The mathworks. Inc., Natick, MA. (1998).

[13] Mo, Yilin, and Bruno Sinopoli. “Secure control against replay attacks.” 47th Annual Allerton Conference on Communication, Control, and Computing, pp. 911–918, 2009.

[14] Mo, Y., Kim, T. H., Brancik, K., Dickinson, D., Lee, H., Perrig, A., and Sinopoli, B. “Cyber-physical security of a smart grid infrastructure.” Proceedings of the IEEE, 100(1), pp. 195–209, 2012.

[15] F. Miao, M. Pajic and G. J. Pappas. “Stochastic Game Approach for Replay Attack Detection.”, University of Pennsylvania, Philadelphia, PA, USA. 2013.

Capacidades de Detección de las Herramientas de Análisis de Vulnerabilidades en Aplicaciones Web

Fernando Román Muñoz, Iván Israel Sabido Cortés, Luis Javier García Villalba

Grupo de Análisis, Seguridad y Sistemas (GASS), Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid
Email: {froman, javiergv}@fdi.ucm.es, isc_86@hotmail.com

***Resumen*—Debido al continuo incremento del número de vulnerabilidades en las aplicaciones Web, se han elaborado diversas clasificaciones para mantener organizadas estas vulnerabilidades, y también se han desarrollado herramientas para detectarlas. Hasta este momento, según nuestra información, no se ha realizado ningún estudio sobre las capacidades de estas herramientas en la detección de las vulnerabilidades presentes en las clasificaciones de vulnerabilidades. En este trabajo mapeamos y agrupamos las diferentes clasificaciones para obtener un conjunto lo más completo posible de vulnerabilidades web, y comprobamos si las herramientas mejor valoradas disponen de las características necesarias para detectarlas. Después introducimos la aplicación web vulnerable que hemos desarrollado, intentando incorporar todas las vulnerabilidades web de nuestra lista, con el objetivo de comprobar las capacidades reales de detección de las herramientas de análisis de vulnerabilidades web.**

Palabras clave*—Clasificaciones de vulnerabilidades, precisión de las herramientas de análisis de vulnerabilidades web, vulnerabilidades web. *(Vulnerability classifications, web vulnerability scanners accuracy, web vulnerabilities).

I. Introducción

La opción más habitual para detectar las vulnerabilidades de una aplicación Web es utilizar una herramientas automática de análisis. A estas herramientas se les proporciona un conjunto de URLs y eventualmente unas credenciales, con las que explorar la aplicación. Una vez que la ha recorrido, intentará introducir determinados valores en los campos y cabeceras, para posteriormente analizar el resultado en busca de evidencias de vulnerabilidades.

Para saber que herramienta de las existentes es mejor en la detección de vulnerabilidades se han realizado varios estudios. En ellos básicamente se parte de un conjunto de herramientas comerciales o de código libre, un conjunto de vulnerabilidades a detectar, y una aplicación con esas vulnerabilidades. Se configuran las herramientas para analizar la aplicación vulnerable y se analizan los resultados. El comparar unos estudios con otros es difícil ya que en cada uno de ellos las herramientas analizadas, el conjunto de vulnerabilidades y la aplicación vulnerable son diferentes.

El conjunto de herramientas disponibles cambia a lo largo del tiempo, por lo que no se puede definir un conjunto estático de herramientas. Pero lo que sí existen son clasificaciones de vulnerabilidades Web que incluyen las categorías existentes de vulnerabilidades. Estas clasificaciones aunque se revisan periódicamente, no están en constante actualización.

En este documento los autores describen el proceso seguido para unificar las clasificaciones disponibles de vulnerabilidades Web, agrupádolas en una única lista. A continuación se revisan las capacidades de detección de las herramientas mejor valoradas frente a las vulnerabilidades de esa lista.

Posteriormente se describe el proceso seguido para desarrollar una aplicación Web vulnerable, intentando que incluya todas las categorías de vulnerabilidades web. Esta aplicación será usada en el futuro para comprobar las capacidades reales de detección de las herramientas de análisis, y para realizar acciones de formación en detección de vulnerabilidades.

II. Antecedentes

Esta sección proporciona información sobre los diferentes conceptos relacionados con las vulnerabilidades y las clasificaciones existentes actualmente.

II-A. Conceptos

Aunque algunas listas de vulnerabilidades clasifican según el concepto de vulnerabilidad, otras lo hacen según otros conceptos como: amenaza, debilidad, riesgo o control. El utilizar diferentes conceptos no impide que puedan compararse las clasificaciones, ya que todos ellos están relacionados, como puede verse en la guía NIST Special Publication 800-30 [1] y en el estándar internaciónal ISO/IEC 27001:2005 [2]. Por motivos de claridad en este documento nos referiremos a todos ellos como vulnerabilidad.

II-B. Herramientas de análisis de vulnerabilidades Web

En un artículo anterior [31] se agrupaban las principales carencias de las herramientas de análisis de vulnerabilidades Web, y se proponían soluciones para varias de ellas. Posteriormente en [4] se proponían mejoras de las soluciones indicadas en ese artículo anterior. En otro artículo anterior [5], se analizaban varios estudios relevantes sobre estas herramientas, tanto comerciales como de código libre. Los resultados muestran las herramientas, vulnerabilidades y aplicaciones vulnerables que se usaron en cada estudio. En lo referente a las herramientas analizadas, aunque en cada estudio se comparan al menos siete herramientas, sólo dos de ellas se valoran en todos los

estudios, y muchas solamente en uno. En lo que respecta a las vulnerabilidades, entre todas se prueban 35, pero sólo las dos más conocidas se pruebas en todos ellos: Cross Side Scripting e inyección SQL. Sobre la aplicación vulnerable utilizada, ninguna se usa en más de un artículo, en algunas se usan aplicaciones desarrolladas a propósito para sus pruebas, y en otras aplicaciones de uso habitual con vulnerabilidades conocidas. Finalmente, en los resultados, ninguna herramienta aparece en las tres primeras posiciones de todos los artículos, y hay muchas de ellas que aparecen en las primeras posiciones de unos y en las últimas de otros. Agrupando los resultados de estos estudios se obtiene una jerarquía de herramientas, en la que que aparecen las siguientes en los primeros puestos: Appscan [20], Acunetix [21], Webinspect [22] y Burp Suite [23].

II-C. Clasificaciones de vulnerabilidades

En esta sección se indican las clasificaciones de vulnerabilidades más relevantes hasta el momento de la elaboración de este documento. Se incluyen clasificaciones tanto de vulnerabilidades en aplicaciones Web, como otro tipo de vulnerabilidades.

La organización Web Application Security Consortium (WASC) proporciona una clasificación de 49 amenazas en aplicaciones Web (WASC TC) [6]. Divide las amenazas entre debilidades y ataques, e incluye la fuente de las amenazas: diseño, implementación o desarrollo. Contiene descripciones y ejemplos. La última versión es la 2.0 liberada en 2010.

La organización Open Web Application Security Project (OWASP) mantiene actualizada la lista OWASP Top 10 [7], que incluye los riesgos de Seguridad más críticos en las aplicaciones Web. Obtiene sus datos de empresas de consultoría y de fabricantes de herramientas de detección de vulnerabilidades. La versión actual es del 2013. Esta lista incluye la descripción y ejemplos de cada vulnerabilidad, y también métodos para mitigar su impacto. OWASP también desarrolla la Guía de Pruebas de OWASP [8] que describe un conjunto de pruebas que se pueden realizar sobre las aplicaciones Web para detectar vulnerabilidades. Contiene 66 pruebas, y la versión actual es la v3 del 2008. Incluye la descripción de las vulnerabilidades, ejemplos y métodos de detección. No se indican métodos para mitigar su impacto, pero sí incluye referencia a otra información de interés de OWASP. La siguiente versión v4 actualmente está en desarrollo.

Del proyecto NIST SAMATE surge en 2007 la guía NIST Special Publication 500-269 [9] que describe las tareas que debe realizar una herramientas de detección de vulnerabilidades Web. En su anexo A se incluye una lista de 14 vulnerabilidades Web que una de estas herramientas debe ser capaz identificar. Las vulnerabilidades de esta lista se han incluido según su probabilidad de ser explotadas. En el anexo B se suguieren brevemente métodos para mitigarlas.

En 2009 la organización WASC también elabora el Web Application Security Scanner Evaluation Criteria (WASSEC) [10], que incluye una lista de problemas de Seguridad que una herramienta de análisis de aplicaciones Web debe detectar. Los extrae principalmente de WASC TC, e incluye 55 problemas agrupados en varias categorías: autenticación, autorización, ataque del lado del cliente, ejecución de comandos y revelación de información.

La clasificación Common Weakness Enumeration (CWE) [11] desarrollada por la Corporación MITRE, es un conjunto de debilidades software en todo tipo de software, no sólo aplicaciones Web. La versión existente en el momento de este documento, la 2.5 de 2013, incluía 940 debilidades. También se citan ejemplos, métodos de detección y de mitigación, y referencias a otras clasificaciones de vulnerabilidades.

SANS y MITRE han desarrollado la clasificación CWE/SANS TOP 25 de los errores software más peligrosos (SANS CWE/25) [12], que fue actualizada por última vez en 2011. Es un subconjunto de CWE e igualmente incluye vulnerabilidades de todo tipo de aplicaciones. Incluye ejemplos y métodos de detección.

Otra clasificación que incluye vulnerabilidades en todo tipo de software es Common Attack Patterns Enumeration and Classification (CAPEC) [3] que mantiene la corporación MITRE. Contiene patrones de ataque y la última version durante la elaboración de este documento es la 2.1 de 2013. Incluye ejemplos y descripciones de flujos de ataque.

Por ultimo Shay Chen mantiene la clasificación SecToolMarket [13] donde se incluye una clasificación de 33 características de auditoria de las herramientas de análisis de aplicaciones web. Se actualiza normalmente cada año e incluye referencias a otras clasificaciones. En Sectoolmarket también se analizan 62 herramientas, para determinar cuales de esas 33 características de auditoría detecta cada una.

II-D. Relaciones entre clasificaciones

En el apartado anterior se indicaban las principales clasificaciones de vulnerabilidades. Para relacionar las vulnerabilidades de unas clasificaciones con otras se han realizado varios trabajos de mapeo. Suelen incluir todas las vulnerabilidades de una clasificación, relacionando las que sean posibles con vulnerabilidades de otras clasificaciones. A continuación se ofrece una breve descripción de los principales mapeos entre clasificaciones.

Denim Group [14] realizó en 2010 un mapeo entre las vulnerabilidades de WASC TC v1.0, SANS CWE/25, y OWASP Top 10 2004 y 2007.

Jeremiah Grossman [15] mapeó en 2009 las clasificaciones WASC TC v2.0 y OWASP Top 10 2010.

Threat Classification Taxonomy Cross Reference View (Webappsec) [16] es una vista de la clasificación WASC actualizada en 2013, que relaciona las vulnerabilidades en WASC TC v2.0 con las de CWE, CAPEC, OWASP Top Ten (2004, 2007 y 2010) y SANS CWE/25. Para ello usan la información de los dos mapeos indicados antes: Denim Group y Jeremiah Grossmans.

La clasificación que incorpora la guía NIST Special Publication 500-269 en su anexo A incluye un mapeo con CWE, OWASP Top 10 2007 y Common Vulnerabilities and Exposures (CVE) [17].

Securing Telligent Evolution [18] es un documento creado por Telligent en 2012, que describe las amenazas que se prueban en la plataforma Telligent Evolution, incluyendo un mapeo entre la guía de pruebas de OWASP y WASC.

Finalmente SecToolMarket incluye relaciones entre las vulnerabilidades de WASC TC v2.0 y la guía de pruebas de OWASP.

La información de todos estos mapeos se puede resumir para determinar qué clasificaciones están más relacionadas. La clasificación WASC TC v2.0 está relacionada con otras 8 en los mapeos analizados, SANS CWE/25 con 5 clasificaciones, OWASP Top 10 2007, OWASP Top 10 2010 y WASC TC v1.0 con tres, y la guía de pruebas de OWASP, CWE, CVE, y CAPEC sólo con otra clasificación.

II-E. Aplicaciones Web vulnerables

Para probar las herramientas existen multitud de aplicaciones Web vulnerables a propósito, algunas de ellas desarrolladas por organizaciones dedicadas a la seguridad, y otras por los fabricantes de las herramientas de detección. Dentro del primer grupo las más relevantes son las siguientes.

Damn Vulnerable Web Application (DVWA) [28] es una aplicación desarrollada en PHP, con vulnerabilidades basadas en la clasificación OWASP Top 10, que ofrece la opción de cambiar el nivel de seguridad. WebGoat [29] es mantenida por OWASP y está basada en tecnología J2EE. Esta aplicación cuenta con más de 30 lecciones para aprender a detectar las vulnerabilidades y corregirlas. OWASP también mantiene Mutillidae [30], desarrollada en PHP, y como DVWA, se basa en la clasificación OWASP Top 10. También tiene distintos niveles de dificultad e incluye información para detectar las vulnerabilidades. Dentro del grupo de aplicaciones vulnerables desarrolladas por los fabricantes de herramientas de detección podemos indicar, a modo de ejemplo, la aplicación vulnerable de Acunetix [27] desarrollada en PHP, o la de IBM AppScan [26] desarrollada en ASP.NET.

En esta sección se han introducidos los conceptos relacionados con las vulnerabilidades, las principales clasificaciones y la relaciones entre ellas, y las aplicaciones vulnerables. En la siguiente sección se explica el proceso seguido para unificar las clasificaciones.

III. Clasificación de vulnerabilidades Web unificada

A partir de las principales clasificaciones de vulnerabilidades y los mapeos entre ellas se puede desarrollar una nueva clasificación que incluya los tipos de vulnerabilidades de todas ellas, sin vulnerabilidades redundantes. Para ello como primer paso nos quedarnos con las clasificaciones completas que sólo incluyen vulnerabilidades Web, es decir no se tienen en cuenta las clasificaciones que incluyen las vulnerabilidades de todos los tipos de aplicaciones, como CWE, ni las que sólo incorporan las vulnerabilidades más relevantes o frecuentes, como hace OWASP Top 10. El resultado son las clasificaciones WASC TC, la guía de pruebas de OWASP y la clasificación de Sectoolmarket. Los mapeos entre ellas son el de Telligent y el de Sectoolmarket. El segundo paso es unificar los mapeos de Telligent y de Sectoolmarket eliminando la información redundante. De esta forma se obtiene un listado de 29 mapeos entre elementos de WASC TC y la guía de pruebas de OWASP. En tercer lugar las vulnerabilidades restantes de cada una de estas clasificaciones se revisan teniendo en cuenta sus descripciones. De esta forma se obtienen 12 nuevas relaciones entre vulnerabilidades de WASC TC y la guía de pruebas de OWASP. Por último se obtiene la clasificación final añadiendo las vulnerabilidades sin relacionar en WASC TC, la guía de pruebas de OWASP y la clasificación de Sectoolmarket. De WASC TC se obtienen ocho, nueve de la guía de pruebas de OWASP, y cinco de la clasificación de Sectoolmarket. Al final tenemos una clasificación de 63 tipos de vulnerabilidades. Esta nueva clasificación (ULWeV) y los mapeos entre las clasificaciones seleccionadas pueden consultase en [19].

En la tabla I puede verse las vulnerabilidades de la nueva clasificación obtenida agrupadas según su procedencia: WASC TC, guía de pruebas OWASP que no están en WASC TC, y Sectoolmarket que no están en las dos anteriores.

Tabla I
Origen de las vulnerabilidad en la nueva clasificación

Clasificación	Vulnerabilidades
WASC TC	Todas (de WASC-01 a WASC-49)
Guía de OWASP	-Information Gathering -User enumeration -Weak Multiple Factors Authentication -Race Conditions vulnerability -DB Listener weak -Code Injection -Cookie attributes -Exposed sensitive session variables -Web services testing
Sectoolmarket	-EL Injection -Padding Oracle -Server Side Java Script (SSJS/NoSQL) Injection -Unrestricted File Upload and Blind/Time-Based SQL Injection -Source Code Disclosure

IV. Precisión de las herramientas de detección de vulnerabilidades Web

En la sección 2 se indicaban los conceptos relacionados con las vulnerabilidades, las herramientas de detección de vulnerabilidades mejor valoradas, las clasificaciones actuales de vulnerabilidades, y las relaciones entre ellas. En la sección 3 se describía el proceso seguido para unificar estas clasificaciones, y obtener una clasificación de vulnerabilidades lo más completa posible. En esta sección se usará esa nueva clasificación de vulnerabilidades para comprobar las características (capacidades de detección) de estas herramientas.

IV-A. Capacidades de detección de las herramientas

Para revisar las capacidades de detección de las herramientas se ha podido conseguir versiones válidas de AppScan, Webinspect y Acunetix. De Burp Suite no se ha podido

conseguir una versión válida completa, por lo que se ha sustituido por Zaproxy [24].

La clasificación ULWeV tiene 63 vulnerabilidades, que se puede agrupar en tres grupos, como se ha visto anteriormente en la tabla I: (1) 49 de WASC TCv2.0, (2) nueve de la guía de pruebas OWASP que no están en WASC TC v2.0, y (3) cinco de la clasificación de Sectoolmarket que no están en las otras dos. Para saber si una herramientas tiene capacidad de detectarlas (incorpora la característica necesaria) se usan varias fuentes de información: el sistema de gestión de vulnerabilidades Threadfix [25] junto con el mapeo entre clasificaciones Webappsec, la información en Sectoolmarket, y finalmente la revisión manual de las características de las herramientas. Threadfix es un Sistema de gestión de vulnerabilidades capaz de consolidar informes de vulnerabilidades provenientes de Acunetix, Appscan, Webinspect, Zaproxy y otras herramientas, relacionando las vulnerabilidades detectadas con CWE.

Como se ha visto en un apartado anterior Webappsec relaciona vulnerabilidades en WASC TC v2.0 con las de CWE. A partir de esta información y la que proporciona Threadfix se pueden determinar las vulnerabilidades de WASC TC v2.0 que detecta cada una de las herramientas seleccionadas, lo que se corresponde con el primer grupo (1) de vulnerabilidades de la nueva clasificación ULWeV. Como también se ha visto en un apartado anterior en Sectoolmarket se analizan 62 herramientas para ver si detectan las vulnerabilidades de su clasificación. De aquí se obtiene si las cinco vulnerabilidades del grupo (3) las detectan las herramientas. Para el grupo (2) de las nueve vulnerabilidades de la guía de pruebas de OWASP, y las que no ha sido posible localizar en los dos pasos anteriores, se han revisado manualmente para comprobar si las herramientas tienen capacidad de detectarlas.

En la tabla II se indica el número de pruebas que puede realizar cada herramienta analizada, así como las que se corresponden con las de ULWeV según su clasificación de origen.

Tabla II
CAPACIDADES DE DETECCIÓN DE LAS HERRAMIENTAS

	Acunetix	Appscan	Webinspect	Zaproxy	Las 4 herramientas
Número aproximado de pruebas	350	2000	4300	49	-
63 vulnerabilidades en ULWeV:					
49 de WASC TC v2.0	28	31	27	49	49
8 de de la guía de OWASP	5	4	3	0	5
5 de Sectoolmarket	5	23	3	0	5
Total	38	37	33	49	59

IV-B. Aplicación Web vulnerable

En apartados anteriores se ha elaborado una clasificación unificada de posibles vulnerabilidades en aplicaciones Web, y después se han revisado las capacidades de las herramientas para detectar esas vulnerabilidades. De esta forma se tiene un conjunto actualizado de vulnerabilidades que deberían de detectar las herramientas, y las que podrían detectar cada una. Pero para poder determinar si realmente una herramientas es capaz de detectar una vulnerabilidad hay que probarla.

Para determinar las capacidades reales de detección de una herramienta de análisis se hace necesario disponer de aplicaciones que tengan esas vulnerabilidades. Para ello se han analizado las aplicaciones vulnerables existentes, de forma que se pueda seleccionar una o varias de ellas, que incorporen el mayor número de vulnerabilidades de la nueva clasificación. El resultado es que DVWA tiene 17 vulnerabilidades de las 63 de la clasificación ULWeV, WebGoat tiene 32, Mutillidae 34, la aplicación vulnerable de Acunetix cuenta con 18 vulnerabilidades de ULWeV, y la de AppScan cuenta con 14. Tomando las que más vulnerabilidades incorpora, con WebGoat y Mutillidae se cubre un total de 39 vulnerabilidades de las 63. Las 14 restantes se analizan para determinar si es posible incorporarlas en alguna de las aplicaciones seleccionadas.

Con la nueva clasificación de vulnerabilidades, y la aplicación, o conjunto de aplicaciones, vulnerables a ellas, se pueden realizar acciones de formación sobre los futuros programadores, de forma que es cubran dos objetivos: enseñar a detectar las vulnerabilidades, para realizar pruebas de penetración sobre las aplicaciones Web; y enseñar a programar de forma segura, evitando en la medida de lo posible desarrollar aplicaciones con vulnerabilidades. En la figura 1 se explica el proceso que se intenta seguir en este documento para mitigar, usando formación y concienciación, el desarrollo de aplicaciones Web con vulnerabilidades.

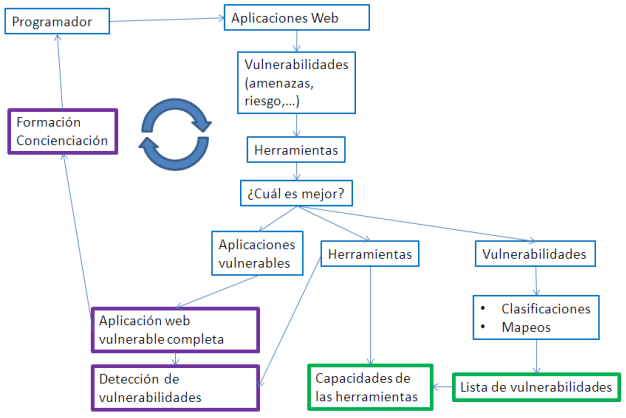


Figura 1. Formación para detectar y mitigar las vulnerabilidades Web

V. ANÁLISIS DE LOS RESULTADOS

Como se ve ninguna de las herramientas analizadas incorpora las capacidades de detección necesarias para detectar las 63

vulnerabilidades, aunque entre las cuatro analizadas sí cubren casi toda la lista de vulnerabilidades.

Aunque Appscan y Webinspect incluyen muchas pruebas de vulnerabilidades, según el análisis realizado son las otras dos Zaproxy y Acunetix las que detectarían más vulnerabilidades deULWeV. Esto es debido a que Appscan y Webinspect incorporan muchas vulnerabilidades de aplicaciones Web específicas y no tanto de tipos de vulnerabilidades. Por ejemplo en las versiones de las herramientas analizadas, Zaproxy incluye solo una característica para detectar Cross-Site Scripting, Acunetix siete, Webinspect más de 500, y Appscan más de 600. En el caso de estas dos últimas, Webinspect y Appscan, solo unas pocas son para detectar la vulnerabilidad en cualquier aplicación, por ejemplo "URL Cross-Site Scripting" la mayoría de ellas son para productos concretos, como "WebSphere Cross-Site Scripting" o "ASP Nuke Cross-Site Scripting Vulnerability". Tanto Sectoolmarket como Threadfix relacionan todas estas vulnerabilidades con el tipo genérico "Cross-Site Scripting". Esto lleva a la conclusión de que herramientas como AppScan o Webinspect son las más adecuadas para analizar aplicaciones cerradas, ya sean comerciales o de software libre, pero que otras herramientas como Acunetix o gratuitas como Zapproxy, deberían de ser al menos igualmente adecuadas para analizar aplicaciones web desarrolladas a medida.

VI. Conclusiones y trabajo futuro

En este artículo se muestra el resultado de unificar las clasificaciones de vulnerabilidades Web más relevantes para obtener una única lista. También se analizan varias de las herramientas mejor valoradas de detección de vulnerabilidades Web, para determinar si incorporan las características de detección necesarias para detectar las vulnerabilidades que incluye la nueva lista. Finalmente se explica la selección de un conjunto de aplicaciones Web vulnerables, y su mejora con vulnerabilidades adicionales, intentando obtener una aplicación que incorpore todas las vulnerabilidades de las clasificaciones. Aunque esta nueva clasificación pueda crecer con el tiempo, sí sirve para determinar que les falta a las herramientas, en que deben mejorar, al dar una visión global de que deben detectar y sus capacidades actuales de detección.

El siguiente paso será analizar estas aplicaciones vulnerables con las herramientas seleccionadas para determinar sus capacidades reales de detección. También esta aplicación vulnerable se podrá utilizar para realizar acciones de formación en desarrollo seguro, al partir de una clasificación completa de tipos de vulnerabilidades, y no solo las más comunes.

Referencias

[1] National Institute of Standards and Technology, "NIST Special Publication 800-30 Revision 1: Guide for Conducting Risk Assessments", 2012.

[2] International Organization for Standardization, "International Standard ISO/IEC 27001", 2005.

[3] The MITRE Corporation, "Common attack patterns Enumeration and Classfication", 2013. Disponible en http://capec.mitre.org/.

[4] F. Román Muñoz, L. J. García Villalba, "Web From Preprocessor for Crawling", en *Multimedia Tools and Applications*, p.p. 1–12, April 2013.

[5] F. Román Muñoz, L. J. García Villalba, "Methods to Test Web Applications Scanners", *Proceedings of the 6th International Conference on Information Technology*, 2013.

[6] Web Application Security Consortium, "The WASC Threat Classification", 2010.

[7] Open Web Application Security Project, "OWASP Top Ten Project", 2013. Disponible en https://code.google.com/p/owasptop10.

[8] Open Web Application Security Project, "OWASP Testing Guide", 2013.

[9] National Institute of Standards and Technology, "Software Assurance Tools: Web Application Security Scanner Functional Specification Version 1.0. NIST Special Publication 500-269", 2008.

[10] Web Application Security Consortium, "Web Application Security Scanner Evaluation Criteria", 2009.

[11] The MITRE Corporation, "Common Weakness Enumeration", 2013. Disponible en http://cwe.mitre.org.

[12] SANS, "CWE/SANS TOP 25 Most Dangerous Software Errors", 2011. Disponible en http://www.sans.org/top25-software-errors.

[13] S. Chen, "General Features Comparison - Web Application Scanners", 2012. Disponible en http://www.sectoolmarket.com.

[14] D. Cornel, "Mapping Between OWASP Top 10 (2004, 2007), WASC 24+2 and SANS CWE/25", 2010.

[15] J. Grossman, "WASC Threat Classification to OWASP Top Ten RC1 Mapping" 2013.

[16] Web Application Security Consortium, "Threat Classification Taxonomy Cross Reference View", 2012.

[17] The MITRE Corporation, "CVE", 2013. Disponible en http://cve.mitre.org.

[18] Telligent, "Telligent Evolution Platform", 2013.

[19] F. Roman, l.-J. Garcia, "Web vulnerability classification mapping", 2013.

[20] IBM, "IBM Security Appscan", 2014. Disponible en http://www-03.ibm.com/software/products/us/en/appscan.

[21] Acunetix, "Acunetix Web Vulnerbility Scanner", 2014. Disponible en http://www.acunetix.com.

[22] HP, "HP Webinspect", 2014.

[23] Portswigger, "Burp Suite",2014. Disponible en http://portswigger.net/burp.

[24] Open Web Application Security Project, "Zaproxy", 2014.Disponible en http://code.google.com/p/zaproxy/.

[25] Denim Group, "Threadfix", 2013. Disponible en https://code.google.com/p/threadfix.

[26] IBM, "IBM Security Appscan", 2014. Disponible en http://demo.testfire.net.

[27] Acunetix, "TEST and Demonstration site for Acunetix Web Vulnerability Scanner", 2014. Disponible en http://testphp.vulnweb.com.

[28] DVWA, "Damn Vulnerable Web Application", 2014. Disponible en http://www.dvwa.co.uk/.

[29] WebGoat, "OWASP WebGoat Project", 2014. Disponible en https://www.owasp.org/index.php/Category:OWASP_WebGoat_Project/.

[30] Mutillidae, "OWASP Mutillidae 2 Project", 2014. Disponible en https://www.owasp.org/index.php/OWASP_Mutillidae_2_Project.

[31] F. Román Muñoz, L. J. García Villalba, "Preproceso de Formularios para el Análisis de Seguridad de las Aplicaciones Web", Actas de la XII Reunión Española sobre Criptología y Seguridad de la Información (RECSI 2012), Donostia-San Sebastián, España, Septiembre 2012.

Sistema de Detección de Atacantes Emascarados Basado en Técnicas de Alineamiento de Secuencias

Jorge Maestre Vidal, Luis Javier García Villalba

Grupo de Análisis, Seguridad y Sistemas (GASS), Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid
Email: jmaestre@ucm.es, javiergv@fdi.ucm.es

***Resumen*—Los ataques enmascarados constituyen la actividad malintencionada perpetrada a partir de robos de identidad, entre la que se incluye la escalada de privilegios o el acceso no autorizados a activos del sistema. Este trabajo propone un sistema de detección de atacantes enmascarados mediante la observación de las secuencias de acciones llevadas a cabo por los usuarios legítimos del sistema. La clasificación de la actividad monitorizada es modelada y clasificada en base a algoritmos de alineamiento de secuencias locales. Para la validación del etiquetado se incorpora la prueba estadística no paramétrica de Mann-Whitney. Esto permite el análisis de secuencias en tiempo real. La experimentación realizada considera los conjuntos de muestras de Schonlau. La tasa de acierto al detectar ataques enmascarados es 98,3 % y la tasa de falsos positivos es 0,77 %.**

Palabras clave*—Atacantes enmascarados, ataques internos, detección de intrusiones, seguridad de la información.** ***(Masquerader attacks, insider attacks, intrusion detection, information security)

I. Introducción

La mayor parte de los accesos no autorizados a un sistema de la información se producen desde el interior, lo que comúnmente se conoce como *ataques internos*. Este tipo de ataques no necesitan explotar vulnerabilidades para atravesar los diferentes controles de acceso ya que de alguna manera, los atacantes satisfacen los requisitos para acceder haciéndose pasar por usuarios legítimos. Los atacantes internos se clasifican en: *traidores* y *enmascarados* [1]. Los *traidores* son usuarios legítimos que ganan privilegios para acceder a información restringida. Los *enmascarados* son atacantes con acceso no autorizado al sistema que suplantan la identidad de usuarios autorizados. Las propuestas para la identificación de atacantes internos varían en función de su clasificación. Dado que los *traidores* son usuarios legítimos, habitualmente conocen las características y la organización del sistema protegido. En consecuencia su detección se centra en la elaboración de trampas y señuelos [2]. Sin embargo la detección de *enmascarados* se centra en la identificación de comportamientos anómalos respecto a los perfiles del uso habitual del sistema. Para ello se tienen en consideración eventos que se producen a nivel local, como los comandos ejecutados, llamadas al sistema, movimientos entre directorios o accesos a ficheros.

El sistema propuesto en este trabajo detecta atacantes enmascarados mediante el análisis de las acciones llevadas a cabo por los usuarios del sistema. Para ello se aplican algoritmos de alineamiento de secuencias, lo que permite modelar su comportamiento e identificar actividades asociadas a su modo de uso indebido. A diferencia de las propuestas anteriores, se introduce el uso de técnicas de alineamiento locales y la prueba de Mann-Whitney para la verificación de sus clasificaciones. Esto permite el análisis en tiempo real de la actividad del usuario sin disminuir la precisión de los procesos de detección. Además cada vez que el usuario ejecuta nuevas acciones su actividad vuelve a ser evaluada. El sistema permite gestionar la emisión de alertas en situaciones dudosas, produciendo una mejora adicional en la etapa de etiquetado.

Este trabajo está estructurado en 7 secciones, siendo la primera de ellas la presente introducción. En la sección II se describen los trabajos previos relacionados con la detección de atacantes enmascarados. En la sección III se explican las principales técnicas de alineamiento de secuencias y sus principales características. En la sección IV se detalla la arquitectura y las características del sistema propuesto. En la sección V se introduce la técnica de validación de etiquetado que complementa al sistema de detección. En la sección VI se describe la experimentación realizada y los resultados obtenidos. Por último, en la sección VII se presentan las conclusiones y las propuestas de trabajo futuro.

II. Trabajos Relacionados

La detección de atacantes enmascarados tiene sus orígenes en propuestas que tienen como objetivo la identificación de comandos poco utilizados por los usuarios legítimos. En [3] se introduce la separación de los comandos ejecutados en usuales e inusuales. La etapa de análisis es llevada a cabo mediante cadenas de Markov: a mayor concentración de comandos inusuales, mayor es la probabilidad de que se trate de un atacante enmascarado. En [4] se propone el estudio de cadenas de acciones mediante el uso de una ventana deslizante de tamaño fijo y la incorporación de un alfabeto de secuencias. Cuando el contenido de la ventana analizada no encaja con ninguna palabra del alfabeto, se gestiona como una cadena extraña, incrementando la posibilidad de que sea etiquetada como una intrusión.

Es importante destacar la aportación de Scholau et al. [5] [6]. [5] presenta la estrategia *Uniqueness*, basada en el análisis de la aparición de comandos que no figuran en los conjuntos de

muestras de referencia. Asimismo, proponen la aplicación de métodos de compresión y la identificación de la naturaleza de las secuencias mediante el análisis de su comportamiento. [6] estudia la precisión de diferentes técnicas de detección, destacando entre ellas *Uniqueness*, factores bayesianos, Markov, compresión o encaje de secuencias. Para llevar a cabo su evaluación aplica por primera vez el *dataset* conocido como SEA. Este conjunto de muestras es aplicado posteriormente por otros autores en la evaluación de propuestas similares. No obstante, tal y como señalaron Maxiom y Townsend, su uso es controvertido [7]. Su crítica se centra en que las capturas de los distintos usuarios están mezcladas, se desconoce la información sobre sus fuentes, se desconocen los períodos de captura y no se da información específica acerca de las tareas que llevaron a cabo los usuarios legítimos durante el proceso de captura. A pesar de ello los conjuntos de muestras de Schonlau son considerados un estándar funcional para la validación de este tipo de sistemas.

Algunos trabajos han propuesto alternativas al estudio de las secuencias de comandos o llamadas al sistema. En [8] la elaboración de los perfiles de usuario considera la frecuencia entre *clics* y otras características de los movimientos del ratón. Asimismo algunas propuestas se centran en el estudio de la actividad de ususarios en redes. Este es el caso de [9], donde sus perfiles se elaboran en base a eventos como búsquedas, descargas, impresiones o movimientos en redes sociales.

[10] es uno de los trabajos de mayor impacto. Su objeto de análisis son los comandos introducidos por los usuarios. Sin embargo, en esta ocasión propone su tratamiento tras un agrupamiento y etiquetado en base a su funcionalidad, como la recopilación de recursos, búsquedas o procesos de comunicaciones. También propone técnicas para la generación de conjuntos de muestras con actividades anómalas. Las muestras de sus experimentos se basan en los resultados de un juego de *captura de bandera* en el que usuarios desconocedores del sistema tratan de localizar un archivo determinado mientras su actividad es monitorizada

En [11] se propone el modelado de eventos del sistema mediante los PHMMs (*Profile Hidden Markov Models*), previamente aplicados en el campo de la bioinformática. Los experimentos realizados arrojan una gran precisión cuando se consideran conjuntos pequeños de muestras para su entrenamiento. En [12] se introduce el concepto de ataque de mimetismo en el contexto de la detección de atacantes enmascarados. Asimismo se demuestra la vulnerabilidad de la mayor parte de las propuestas actuales frente a estrategia de evasión similares y se proponen algoritmos para su mitigación. En [13] se introducen las técnicas de alineamiento de secuencias para el análisis de las acciones llevadas a cabo por los usuarios del sistema. Entre su contenido destaca la discusión sobre la aplicación de las diferentes técnicas de alineamiento, el diseño de técnicas para la actualización en tiempo real de los diccionarios de referencia y la propuesta de diferentes sistemas de puntuación. Los algoritmos implementados implica un alto consumo de recursos computacionales. En consecuencia se proponen heurísticas para reducir su consumo a costa de penalizar la precisión de la etapa de análisis.

III. Alineamiento de Secuencias

El alineamiento de secuencias es una técnica procedente del campo de la bioinformática que tiene como finalidad establecer el grado de similitud entre cadenas de ADN, ARN o diferentes proteínas. Las secuencias alineadas generalmente corresponden a nucleótidos o aminoácidos, y se identifican mediante símbolos de un alfabeto. Cuando el ancestro de un linaje de individuos es común, las diferencias son consideradas mutaciones puntuales (sustituciones). A los huecos se los denomina *indels* (inserciones o eliminaciones). La similitud es estudiada como una medida de conservación entre linajes, que habitualmente conlleva una importancia funcional y estructural de las muestras.

Las diferentes técnicas de alineamiento de secuencias habitualmente son consideradas como una generalización del problema de la detección de la longitud de la sub-secuencia más larga común entre dos cadenas, conocida como LCS (*Longest Common Subsequence*). Para la obtención de la LCS el proceso de alineamiento consiste en la eliminación de símbolos y en añadir huecos (*gaps*) hasta que aparezcan sub-secuencias similares, determinándose las de mayor dimensión. Las estrategias de alineamiento de secuencias se clasifican en función de su objeto de análisis. Cuando las secuencias son alineadas considerando su extensión total reciben el nombre de alineamiento global. Si lo hace considerando sus diferentes sub-secuencias reciben el nombre de alineamiento local. Finalmente, cuando se combinan de tal manera que se considera la similitud de una secuencia completa respecto a sub-secuencias de otra secuencia diferente, reciben el nombre de alineamiento semi-global.

IV. Sistema de Detección de Atacantes Enmascarados

El sistema de detección propuesto construye secuencias de acciones llevadas a cabo por los usuarios del sistema a nivel de eventos, y aplica el algoritmo de Smith-Waterman [14] para su alineamiento local. El análisis de sub-secuencias permite la identificación de aquellas situaciones en las que las acciones efectuadas por el atacante se mezclan con las del usuario legítimo: por ejemplo, cuando el usuario con acceso autorizado abandona su puesto de trabajo sin cerrar sesión. En este caso el atacante puede aprovechar su ausencia para efectuar actividades maliciosas, retirándose antes de ser detectado.

En la Figura 1 se muestra su arquitectura. El proceso de detección se lleva a cabo de la siguiente manera: una vez comenzada la monitorización de las acciones realizadas por un usuario, cada vez que se ejecuta una nueva acción es incluida en la cadena *Test*. El proceso de análisis consiste en alinear *Test* con cada una de las secuencias de la colección $Legit = l_1, l_2, .., l_m$, las cuales contienen acciones realizadas habitualmente por usuarios legítimos del sistema. Para contrastar los resultados, *Test* también es alineada con las secuencias de la colección $Intrusions = I_1, I_2, .., I_p$, las cuales contienen actividades anómalas perpetradas por

usuarios no familiarizados con el sistema. Una vez establecidas las puntuaciones se aplica la prueba de validación. Cuando el nivel de parecido de *Test* con alguna de las colecciones es lo suficientemente representativo, es etiquetada acorde a las características de sus secuencias El sistema propuesto repite este proceso por cada nueva acción monitorizada, y solo se detiene si la prueba de validación es superada. De lo contrario, se sobrentiende que no se dispone de suficiente información para decidir la naturaleza del usuario, y se espera a que se ejecuten nuevas acciones.

El sistema de puntuaciones para el algoritmo de alineamiento de secuencias añade $+3$ a la puntuación final en los casos en que el valor de una posición de *Test* conincide con el de su posición análoga en alguna secuencia de las colecciones de referencia. Sin embargo, en caso de incoherencia no se producen modificaciones. La penalización por *gap* en las secuencias de las colecciones es de -2 mientras que en la cadena Test es de -3. A pesar de ello, la puntuación mínima emitible (menor nivel de similitud) es 0.

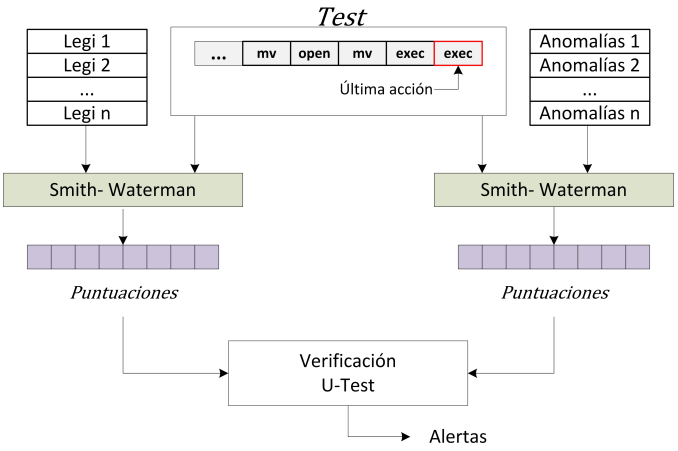


Figura 1. Arquitectura del sistema de detección

V. Validación del Etiquetado

La validación del etiquetado se realiza mediante la prueba de Mann-Whitney, conocida como U-test. Se trata de una extensión del T-test de Student no paramétrica adaptada al análisis de dos muestras independientes. Su objetivo es la comprobación de que dos muestras simétricas han sido extraídas a partir de la misma población. Para ello los datos deben de estar medidos en una escala ordinal, lo que implica la necesidad de ordenar las puntuaciones obtenidas. El cálculo del estadístico U, parte de los valores U_1 y U_2, y es definido como $U = min\{U_1, U_2\}$ donde

$$U_1 = n_1 n_2 \frac{n_1(n_1+1)}{2} - R_1 \quad (1)$$

$$U_2 = n_1 n_2 \frac{n_2(n_2+1)}{2} - R_2 \quad (2)$$

Siendo n_1 y n_2 las longitudes de los vectores de las puntuaciones ordenadas y R_1 y R_2 las sumas de los rangos de las observaciones de las muestras. Dado que el número de muestras es grande, U tiende a parecerse a la distribución normal, de manera que

$$z = \frac{U - M_u}{\sigma_u} \quad (3)$$

Donde m_u y σ_u son la media y la desviación estándar de U, formuladas de la siguiente manera

$$m_u = \frac{n_1 n_2}{2} \quad (4)$$

$$\sigma_u = \frac{n_1 n_2 (n_1 + n_2 + 1)}{12} \quad (5)$$

Una vez obtenida la probabilidad de pertenencia, se comprueba que la cota de porcentaje de error sea admisible. En ese caso puede confirmarse que la diferencia entre las puntuaciones de *Test* con las colecciones de secuencias es considerable. El siguiente paso es decidir con cuál de ellas presenta un mayor parecido. Para ello se considera la media de las puntuaciones obtenidas con cada grupo. La secuencia de puntuaciones cuyo promedio es más bajo corresponde con el conjunto de mayor similitud. Cuando *Test* presenta un mayor parecido con la colección de secuencias legítimas, el usuario se etiqueta como legítimo. Si lo hace con la colección de secuencias anómalas es etiquetado como atacante enmascarado. Si no hay una diferencia clara, la prueba concluye de manera indeterminada: no se tiene suficiente información para establecer un etiquetado preciso. En ese caso se añaden nuevas acciones a *Test* y se repite el proceso hasta que la prueba de validación es superada.

VI. Evaluación del Sistema

Para llevar a cabo los experimentos se ha empleado la colección de muestras de Schonlau [6]. Los datasets de Schonlau están compuestos por capturas de las actividades realizadas por 50 usuarios distintos operando sobre entorno Unix en el año 1998. A pesar de su antigüedad son consideradas un estándar funcional para la evaluación de sistemas de detección de ataques enmascarados. Están organizados de manera que a cada usuario le corresponde un fichero que contiene una serie de 15,000 acciones llevadas a cabo durante el periodo de captura. Los primeros 5,000 comandos corresponden a actividades legítimas, por lo que han sido utilizadas en la elaboración de la colección de secuencias de actividades legítimas. Los siguientes 10,000 comandos pueden tratarse tanto ataques de enmascaramiento, como de actividades legítimas. Se ha extraído parte de los ataques enmascarados para la elaboración de la colección de actividades anómalas.

La evaluación del sistema de detección consiste en un proceso de validación cruzada que involucra los distintos usuarios y los ataques enmascarados presentes en las colecciones de Schonlau. La Tabla I muestra la tasas de falsos positivos o TPR (*True Positive Ratio*) y la tasa de falsos positivos o FPR (*False Positive Ratio*) obtenidos al determinar diferentes longitudes en las secuencias que componen la colección de actividades legítimas de referencia.

Tabla I
TPR/FPR EN FUNCIÓN DE LA LONGITUD DE SECUENCIA

Prec.\Long.	10	20	30	40	100	200	400	600	800
TPR	0.96	0.97	0.96	0.960	0.97	0.983	0.974	0.9701	0.9712
FPR	0.02	0.02	0.02	0.01	0.01	0.0077	0.0172	0.0232	0.0276

El mejor resultado se obtiene con secuencias de longitud 200, siendo $TPR = 98{,}3\,\%$ y F$PR = 0{,}77\,\%$ Sin embargo el peor resultado se obtiene cuando la longitud es 10, con $TPR = 96{,}9\,\%$ y $FPR = 2{,}68\,\%$. Los cambios más representativos se encuentran en las variaciones de FPR. La curva que describe estos cambios muestra un punto de inflexión al considerarse longitud 200, siendo este su valor mínimo.
El análisis por separado del comportamiento de los usuarios del sistema con secuencias de longitud 200 indica que existen usuarios más propensos a ser suplantados con éxito que otros.

En la Figura 2 se muestra la representación en el espacio ROC (*Receiver Operating Characteristic*) de la precisión obtenida en cada uno de ellos. El eje Y del espacio ROC lo constituyen los valores FPR de los experimentos, mientras que el eje X contiene el valor de los FPR. Esto habitualmente es interpretado como el intercambio entre los beneficios obtenidos (TPR) por el sistema, y los costes que conlleva (FPR). La ubicación óptima en el espacio ROC es la esquina superior izquierda con $TPR = 1, FPR = 0$. A este punto se le llama *clasificación perfecta*, y su proximidad determina la calidad de la precisión del sistema.
El valor TPR ha oscilado en el intervalo aproximado del $98 \pm 2\,\%$. El FPR ha variado en el intervalo aproximado $5 \pm 5\,\%$, lo que indica una desviación algo más representativa. La clasificación perfecta ha sido alcanzada por 22 de los 50 usuarios que han participado en la prueba.
A la vista de los resultados arrojados en los experimentos queda demostrada la gran capacidad del sistema propuesto de identificar atacantes enmascarados.

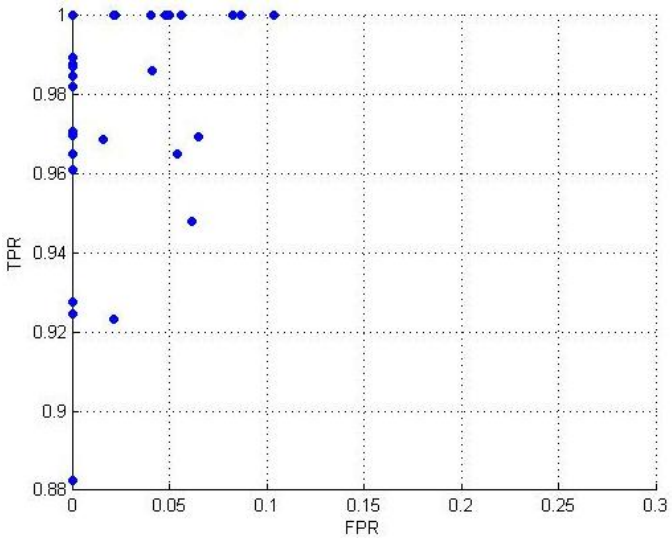


Figura 2. TPR/FPR de cada usuario en el espacio ROC

VII. CONCLUSIONES

Se ha propuesto un sistema para el análisis de la actividad realizada por los usuarios del sistema en tiempo real e identificar atacantes enmascarados. En su evaluación a partir de los conjuntos de muestras de Schonlau ha alcanzado valores promedios de $TPR = 98{,}3\,\%$ y $FPR = 0{,}77\,\%$, lo que demuestra un alto grado de precisión. Sin embargo los experimentos indican que el sistema es sensible a cambios en la longitud de las secuencias que componen las colecciones de actividades legítimas y anómalas. Asimismo se ha comprobado cómo la capacidad de acierto en el etiquetado es mejor en algunos usuarios que en otros. Como trabajo futuro se propone la aplicación de técnicas para homogeneizar la calidad de los conjuntos de entrenamiento y fortalecer el sistema frente a ataques de evasión.

REFERENCIAS

[1] M. B. Salem, S. Hershkop, S. J. Stolfo, "A Survey of Insider Attack Detection Research," *Insider Attack and Cyber Security. Advances in Information Security*, Vol. 39, pp. 69-90, 2008.
[2] M. B. Salem, S. J. Stolfo, "Decoy document deployment for effective masquerade attack detection," *Proceedings of the 8th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA)*, Amsterdam, The Netherlands. Lecture Notes in Computer Science, Vol. 6739 pp. 35-54, July 2011.
[3] H. W. Ju, Y. Vardi, "A hybrid high-order Markov chain model for computer intrusion detection," *Journal of Computational and Graphical Statistics*, Amsterdam, The Netherlands. Lecture Notes in Computer Science Vol. 10(2), pp. 277-295, 2001.
[4] K. Tan, A. Roy, "Why 6?: Defining the operational limits of stide, an anomaly-based intrusion detector," *Proceedings of the IEEE Symposium on Security and Privacy*, Berkeley, CA, USA, pp. 188-201, May 2002.
[5] M. Schonlau, M. Theus, "Detecting masquerades in intrusion detection based on unpopular commands," *Information Processing Letters*, Vol. 76, pp. 33-38, November 2000.
[6] M. Schonlau, W. DuMouchel, W. H. Ju, A. F. Karr, M. Theus, Y. Vardi, "Computer intrusion: Detecting masquerades," *Statistical Science*, Vol. 16, No.1, pp. 58-74, February 2001.
[7] R. A. Maxion, T. N. Townsend, "Masquerade detection using truncated command lines," *Proceedings of the IEEE International Conference on Dependable Systems and Networks (DSN)*, Bethesda, MD, USA, pp. 219-228, June 2002.
[8] A. Garg, R. Rahalkar, S. Upadhyaya, K. Kwiat, "Profiling users in GUI based systems for masquerade detection," *Proceedings of the IEEE Information Assurance Workshop (IAW)*, West Point, NY, USA, pp. 48-54, June 2006.
[9] M. A. Maloof, G. D. Stephens, "Elicit: A system for detecting insiders who violate need-to-know," *Proceedings of the 10th International Symposium on Recent Advances in Intrusion Detection (RAID)*, Gold Goast, Australia. Lecture Notes in Computer Science, Vol. 4637, pp. 146-166, September 2007.
[10] M. B. Salem, S. J. Stolfo, "Modeling User Search Behavior for Masquerade Detection," *Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection (RAID)*, Menlo Park, CA, USA. Lecture Notes in Computer Science, Vol. 6961, pp. 181-200, September 2011.
[11] L. Huang, M. Stamp, "Masquerade detection using profile hidden Markov models," *Computers & Security*, Vol. 30 (8), pp. 732-747, November 2011.
[12] J. E. Tapiador, J. A. Clark, "Masquerade mimicry attack detection: A randomised approach," *Computers & Security*, Vol. 30 (5), pp. 297-310, May 2011.
[13] S. Coull, B. Szymanski, "Sequence alignment for masquerade detection," *Computational Statistics & Data Analysis*, Vol. 52 (8), pp. 4116-4131, April 2008.
[14] T. F. Smith, M. S. Waterman, "Identification of common molecular subsequences," *Journal of Molecular Biology*, Vol. 147 (1), pp. 195-197, 1981.

Cadena de Custodia en el Análisis Forense. Implementación de un Marco de Gestión de la Evidencia Digital.

Tomás Marqués-Arpa
Estudios de Informática, Multimedia
y Telecomunicación.
Universitat Oberta de Catalunya.
Email: tomasmarques@uoc.edu

Jordi Serra-Ruiz
Estudios de Informática, Multimedia
y Telecomunicación.
Universitat Oberta de Catalunya.
Email: jserrai@uoc.edu

***Resumen*—Uno de los problemas principales en el análisis forense de la información es la CdC (Cadena de Custodia), es decir, el procedimiento de trazabilidad de todas las pruebas que se obtienen durante las distintas etapas del proceso de instrucción judicial. Generalmente, las evidencias son obtenidas por los Cuerpos y Fuerzas de Seguridad del Estado y, posteriormente, son examinadas y analizadas por analistas forenses en seguridad de la información. Es imprescindible que la transferencia de información entre las partes implicadas en el proceso se lleve a cabo con las máximas garantías, tanto judiciales como procesales.**

El propósito de este artículo es la propuesta de una CdC digital y segura, vista como un conjunto de eslabones. Gracias a ello, la prueba no perderá valor jurídico puesto que aunque se haya roto un eslabón, habrá quedado asegurada por la solidez de los eslabones anteriores.

***Palabras clave*—Cadena (*chain*), cifrado (*cipher*), custodia (*custody*), estampado (*stamping*), evidencia (*evidence*), forense (*forensic*), geolocalización (*geolocation*), huella (*footprint*), paquete (*package*).**

I. Introducción

En la actualidad, con el auge de las TICs (Tecnologías de la Información y las Comunicaciones), es necesario proporcionar herramientas, métodos y procedimientos que aseguren la misma seguridad para las evidencias digitales.

El estudio se ha desarrollado en virtud de las "líneas de investigación de la Comisión Europea para 2013" [1].

La metodología utilizada ha sido la de "Diseño y Creación" (sensibilización, sugerencia, desarrollo, evaluación y conclusión) [2], [3]. Así, podemos indicar que los principales beneficios o resultados de este estudio en las evidencias digitales, serán los siguientes: la generación de una propuesta en el proceso de creación y transmisión, la contribución para la mejora en la gestión y el planteamiento de un método seguro para el envío.

Las principales cuestiones planteadas son si es posible desarrollar un nuevo método para la CdC dgital, si se puede implementar y en caso afirmativo, si se puede extender a los enlaces, datos y aplicaciones. Otras cuestiones son si el método propuesto es más seguro que el utilizado en la actualidad y con menor carga computacional, así como si existen métodos anteriores similares al tema tratado.

Así pues, se fijan los objetivos de la investigación que son:

- Una revisión, análisis y evaluación de la literatura propuesta [4]–[8].
- La implementación de un MGED (Marco de Gestión de la Evidencia Digital) y el estudio de su funcionalidad.

Análisis forense. En cuanto a la etimología de la palabra forense, se puede decir que viene del latín *forensis* ("antes del foro"), aunque en la actualidad se refiere a algo relacionado con los "Tribunales de Justicia" [9].

Como se define por Clint et al [10] y Carrier [11], la ciencia forense digital es una rama de la ciencia forense que abarca la recuperación e investigación de los materiales que se encuentran en los dispositivos digitales o generados por ellos y a menudo, en relación con delitos informáticos. En la ciencia forense, los principios científicos, métodos y técnicas se aplican a la justicia buscando el bien de la sociedad y de la seguridad pública [9]. Así pues, el forense informático es responsable de asegurar, identificar, preservar, analizar y presentar pruebas digitales de modo que se acepten en los procesos judiciales [9].

Evidencia. Se denomina así a cualquier elemento que proporcione la información, mediante el cual se pueda deducir alguna conclusión o que constituya un hallazgo relacionado con el hecho que está bajo investigación [9].

Cadena de Custodia. Consiste en un informe detallado que documenta la manipulación y el acceso a las pruebas objeto de la investigación. La información contenida en el documento debe ser conservada adecuadamente y mostrará los datos específicos, en particular todos los accesos con fecha y hora determinada [12].

Citando a Colquitt: *"El objetivo pues, de establecer una Cadena de Custodia es para convencer al Tribunal de Justicia de que es razonablemente probable que la exposición sea auténtica y que nadie ha alterado o manipulado la prueba física"* [13].

El Instituto Nacional de Justicia de los EE.UU., define la CdC como *"un proceso que se utiliza para mantener y documentar la historia cronológica de las pruebas"*. Esto significa el control de las personas que recogen la evidencia y de cada persona o entidad que posteriormente tiene la custodia de la misma, de las fechas en las que los artículos fueron recogidos o transferidos, de la agencia y el número del caso o el nombre del sospechoso, así como una breve descripción de cada elemento [14].

En lo que respecta al tratamiento de la evidencia digital en la CdC, podemos citar la norma: "BS 10008:2008. *Especificación sobre las pruebas y admisibilidad legal de la información electrónica*, BSI British Standard" [15]. En ella se incluyen los diferentes aspectos relacionados con el tratamiento de las principales pruebas digitales.

Para probar la CdC, es necesario conocer todos los detalles sobre cómo se manejó la evidencia en cada paso del camino. La vieja fórmula utilizada por la policía, los periodistas y los investigadores de "quién, qué, cuándo, dónde, por qué y cómo" (del inglés "las cinco Ws y una H"), se puede aplicar para ayudar en la investigación forense de la información [7], [16].

Para garantizar la admisibilidad de las pruebas, es necesario prestar especial atención a los métodos y procedimientos utilizados para la obtención de las mismas, respetando no sólo los procedimientos técnicos sino también la legislación judicial y la legislación aplicable al caso. Las medidas tomadas no deben modificar las pruebas y todas las personas involucradas deben ser competentes en procedimientos forenses. Todas las actividades realizadas deben documentarse y conservarse las pruebas, de modo que estén disponibles para la repetición de exámenes con el mismo resultado. En ciertos momentos, los procedimientos podrán llevarse a cabo en presencia de un notario o secretario judicial. Las personas que están a cargo de las pruebas digitales son las responsables de las medidas adoptadas con respecto a ellas mientras estén bajo su custodia [15].

II. Estado del Arte

Marco de Gestión de la Evidencia Digital (MGED). Ćosić y Bača han propuesto el *Digital Evidence Management Framework* [7], mediante el cual es posible desarrollar un marco de gestión sencillo para el proceso de la investigación digital basado en las causas y en los efectos producidos por los eventos. Las fases se pueden organizar en función de los requisitos básicos de la investigación, es decir, habrá que encontrar la evidencia que muestre las causas y efectos de un evento y por tanto, será necesario desarrollar hipótesis sobre los hechos ocurridos en la escena del delito. Cada fase tiene un objetivo claro y los requisitos y procedimientos se pueden desarrollar en consecuencia. Como afirman Carrier y Spafford, se deberán perfilar claramente las definiciones y los conceptos que se utilicen en este marco [17].

En la Figura 1 se muestra la propuesta del concepto del MGED, que garantiza la seguridad de una cadena de custodia sobre la base de los "cinco Ws y una H" que proponen Ćosić y Bača [7]. Aconsejan utilizar una función SHA-2 (*Secure Hash Algorithm*) de la huella digital de la evidencia, una característica biométrica de autenticación e identificación para la firma digital (quién), control de fecha y hora mediante la adición de un estampado generado por una entidad de confianza (cuándo), la utilización de servicios (posicionamiento global por *GPS* y *GLONASS* y/o *Google Maps*) o algún dispositivo de *RFID* para la geolocalización (dónde) y el cifrado asimétrico para asegurar la evidencia digital (cómo).

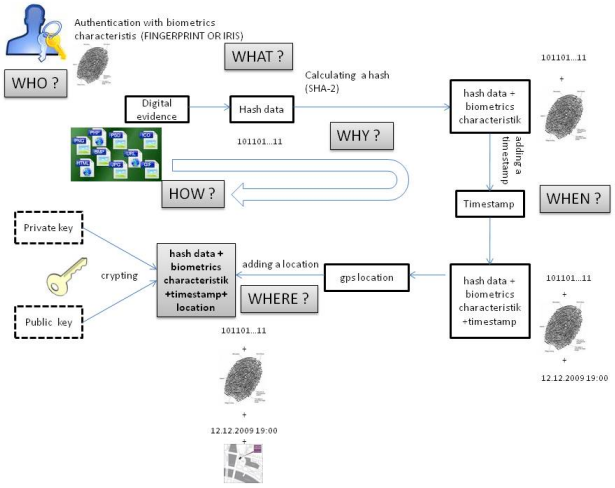


Fig. 1. MGED propuesto por Ćosić and Bača [7].

Huella de la evidencia. Ćosić y Bača proponen que no se utilice la evidencia digital original, en su lugar recomiendan que se maneje una huella digital de las pruebas [7]. Para calcular la huella digital se utilizará una función hash SHA-2, en lugar de las funciones SHA-0 ó SHA-1. Esto se hace para evitar un ataque criptográfico (colisión y/o ataque preimagen).

No hay límite del tamaño del archivo de evidencia digital para el que se desea calcular un hash. Se puede utilizar un archivo (*jpg, tiff, txt,* etc.), un grupo de archivos o algún tipo de archivo específico (*zip, rar, tar,* etc.) o incluso una unidad física (disco duro, memoria externa,etc). Al utilizar una función hash SHA-2, se dará un valor de tamaño fijo (224, 256, 384 ó 512 bits dependiendo de sí se usa SHA-224, SHA-256, SHA-384 ó SHA-512). Las huellas más utilizadas son SHA-256 y SHA-512.

Características biométricas. Ćosić y Bača plantean, con el fin de realizar la autenticación e identificar y conocer a las personas que manejan la evidencia, la utilización de las características biométricas del individuo [7]. Como pueden ser la huella de algún dedo de la mano, las características del iris del ojo, las características morfológicas de la cara, etc. El requisito previo para poder utilizar las características biométricas, es la necesidad de disponer de una base de datos de todas las personas que manejan las evidencias, entre las que se deben incluir los agentes de policía relacionados de alguna manera con el caso, los investigadores que han obtenido

las pruebas de campo, los investigadores forenses, los peritos judiciales y el personal judicial.

Estampado de tiempo. Ćosić y Bača recomiendan para conocer el momento en el tiempo en el que se descubre la evidencia y han sucedido los acontecimientos y acciones, una estampación digital del tiempo utilizando una fuente de confianza conocida [7].

Otros autores como Willassen [18], indican que también es posible el uso de métodos correlativos de sello de tiempo almacenado en el sistema de adquisición y que ya fueron creados por otros sistemas (por ejemplo, mediante la fecha y hora de páginas web generadas dinámicamente).

Gayed et al [4] citan la "web semántica" como solución flexible para simbolizar la diferente información, ya que proporciona los lenguajes de marcas semánticas (*markup*) para la representación de los datos con el apoyo de diferentes vocabularios. Estas características pueden ser explotadas para mostrar el documento tangible de la CdC que asegura su fiabilidad e integridad. Por otra parte, pueden incluirse también los mecanismos de consulta de los datos representados para responder a diferentes cuestiones forenses y de procedencia, formuladas por los jurados sobre el caso tratado.

Ćosić y Bača proponen que el método para esta fase sea un "tiempo de estampado de confianza" [8]. El estándar "RFC 3161" define que la marca de tiempo de confianza es un sello de tiempo emitido por una Autoridad de Certificación (*Trusted Third Party, TTP*), que actúa como una Autoridad de Sellado de Tiempo (*Time Stamping Authority, TSA*) [19]. Cuando se obtiene la evidencia digital, el marco de gestión envía una solicitud a la *TSA* para obtener un certificado de sello de tiempo de confianza. En este proceso hay que tener un acceso al sistema de gestión de la *TSA*, o podemos desarrollar un sistema interno con la infraestructura de la *TSA*. Es imprescindible mencionar que en este tipo de "sistema de tiempo" deben existir unos "auditores externos" que actúan como testigos [7].

Geolocalización. Ćosić y Bača indican que se debe determinar el lugar exacto donde se maneja la evidencia digital y dónde se ha manipulado [7]. Actualmente en los EE.UU. algunos organismos utilizan la tecnología de *RFID* (*Radio Frequency IDentification*), para hacer un seguimiento de la evidencia durante su ciclo de vida. A pesar de que con *RFID* se puede hacer un seguimiento de una evidencia digital, no se pueden conseguir las coordenadas (localización). Por este motivo, otros autores como Strawn [20], recomiendan el uso de un Sistema de Posicionamiento (*GPS* o *GLONASS*) para efectuar la recogida e investigación de las evidencias.

Respecto a la utilización de etiquetas *RFID*, podemos asegurar que es muy práctica en la clasificación y almacenamiento de la evidencia física, como por ejemplo en los depósitos judiciales, porque si se pierde el documento de control es posible encontrar la evidencia. Pero lo ideal es que la evidencia digital incorpore los datos de geolocalización en los metadatos, tal y como se propone en el presente trabajo.

Cifrado asimétrico. Para una seguridad mayor, Ćosić y Bača se refieren a un cifrado asimétrico [7]. La evidencia digital y el valor obtenido se cifrarán con la clave privada recibida de la Autoridad de Certificación y se almacena para su uso posterior. Todo el proceso se representa en la Figura 1.

III. Nuestra propuesta

Propuesta de creación y transmisión de la evidencia digital. Se muestra en la Figura 2 y se basa en el método de los "Cinco Ws y una H" [7], [16].

Ćosić y Bača [7] proponen el uso de la identificación biométrica de la persona que se encarga de la captación de las pruebas, como la mejor forma de referencia. Aunque en las aplicaciones de *Smartphones* su uso está limitado, en la actualidad, se está comenzando a crear aplicaciones para *Android* que detectan el iris del ojo o incluso la huella dactilar en la identificación personal y su posterior uso como medio de pago. Así, en un futuro próximo no será necesario el *PIN* (*Personal Identification Number*) para desbloquear los sistemas como hasta ahora y se aplicará en su lugar la identificación biométrica.

En la identificación sí es posible aplicar el número *IMEI* (*International Mobile Equipment Identity*) del teléfono, así como el número de teléfono asociado a la tarjeta *SIM* (*Subscriber Identity Module*). Debido a la legislación antiterrorista aplicada en la mayoría de los países, los números de teléfono asociados a las tarjetas *SIM* identificarán al propietario.

Para determinar el lugar donde se genera la evidencia digital es necesario el uso de la geolocalización. Para ello, la forma más precisa es mediante el uso de satélites. Hasta hace poco sólo era posible utilizar la constelación de satélites norteamericanos *GPS*, pero a partir de los últimos años también se puede utilizar en combinación los rusos *GLONASS* y, en un futuro próximo, también se podrá utilizar la constelación europea *Galileo* o *GNSS* (*Global Navigation Satellite System*). Si en la actualidad la identificación de la posición se realiza con un error máximo entre 2 y 3 metros, próximamente gracias a la exactitud será de centímetros. Así mismo, el uso de datos cifrados *GNSS PRS* (*Public Regulated Service*) en la geolocalización por parte de los investigadores policiales podrá evitar la posibilidad de ataques *jamming* y *spoofing* mediante interferencias.

La utilización de redes *WiFi* será limitada a *WiFi WPA2 PSK* con clave robusta no contenida en diccionario, que junto a la utilización de redes de telefonía 3G/4G podrán proporcionar geolocalización *"indoor"* mediante el servicio de Google, incluso como verificación de que la localización *"outdoor"* por satélite no está siendo atacada, dentro de los márgenes lógicos de inexactitud del servicio de Google.

Además, existe otra posibilidad de asegurar la geolocalización de las pruebas. Si el dispositivo móvil está conectado a una red de telefonía *GSM*, el proveedor de servicios tiene un registro de las conexiones entre el dispositivo y las antenas en la zona, por tanto el dispositivo está geolocalizado. El problema del uso de estos datos está en que es necesaria

una orden judicial para que el proveedor de servicio de datos telefónicos pueda facilitar la información en una investigación (solicitud de prueba anticipada) [21].

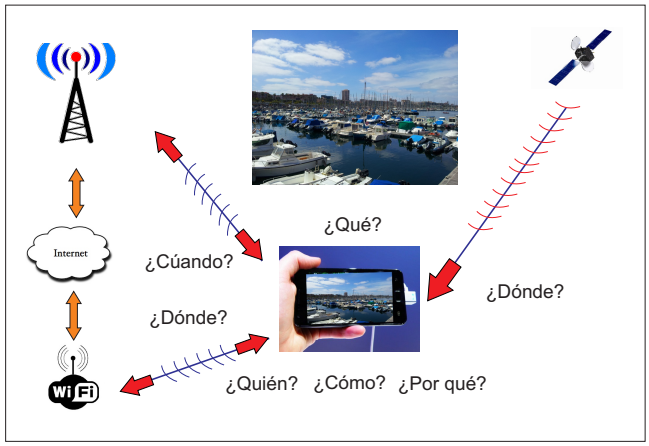


Fig. 2. Propuesta de creación y transmisión de la evidencia digital.

Para generar la evidencia, tal como se muestra en la Figura 2, en primer lugar la cámara debe estar activada en el dispositivo. De este modo se obtiene una fotografía por una persona identificada y cualificada (tanto a nivel técnico como jurídico, normalmente un miembro de las Fuerzas de Seguridad del Estado), para así poder obtener unas pruebas válidas que podrán ser utilizadas en las actuaciones judiciales posteriores. En caso de que el dispositivo se encuentre bajo el área de cobertura de los satélites o con acceso seguro a Internet, la prueba obtenida podrá ser geolocalizada. Una vez obtenida la evidencia con sus respectivos metadatos (datos asociados), se obtiene una huella digital que es enviada de manera segura (con cifrado *SSL/TLS* por el puerto 443) a una *TSA*, la cual devuelve otro archivo (por el mismo enlace seguro) con el "tiempo de confianza de estampado" como está definido en el estándar "RFC 3161", junto con la evidencia que indica la certificación, mediante la fecha y hora de envío.

Propuesta de creación del paquete de evidencia. El paquete estará formado por un archivo *zip* en cuyo interior contendrá los ficheros de evidencias (fotografías, audios y videos), los ficheros devueltos por la *TSA* (en formato *p7s*) y el fichero "documento de pruebas y control de cambios".

Con el fin de garantizar la CdC, es esencial mantener copias de seguridad tanto del paquete de evidencia recibido como del enviado, en dispositivos físicos externos. Así, en caso necesario y a requerimiento de los investigadores forenses, será posible determinar el punto de ruptura de la cadena de custodia y el momento a partir del cual la evidencia deja de ser válida, pero se evita su anulación.

El diseño de la CdC debería ser genérico y no debe limitarse al tamaño de las evidencias, cuyo valor puede ser desde unos pocos *MBytes* (fotografías, audios, etc.) hasta valores de *TBytes* (discos duros). Aunque para ficheros pequeños es posible su transmisión por correo eléctrónico, la forma más segura de envío es a través de un servidor *SFTP* (*Secure File Transfer Protocole*) o un *FTPS* (*FTP-SSL*) que proporcionen acceso remoto y, sobre todo, seguro. Aunque en los dos protocolos se recurre al algoritmo asimétrico (*RSA*, *DSA*), algoritmo simétrico (*AES*), y un algoritmo de intercambio de claves, para la autenticación del *FTPS* utiliza certificados X.509, mientras que *SFTP* utiliza las claves *SSH*. Por otro lado, aunque *SFTP* es más avanzado que *FTPS*, algunos dispositivos pueden no ser compatibles con *SFTP* (como los móviles, consolas etc) y sin embargo con *FTPS* sí lo son.

La posibilidad de que la evidencia sea interceptada (*phishing*, ataques al servidor, etc) hace que sea muy conveniente su cifrado, por lo que se propone *AES* 128 o, preferiblemente, 256 bits [22], [23].

Mediante una herramienta alojada en la Web segura (para prevenir ataques *wiretapping* y *man-in-the-middle*) de la compañía *DigiStamp* que actúa como *TSA* (*https://www.digistamp.com*), se obtienen las huellas digitales (SHA-2, 256 ó 512bits). La *TSA* crea un archivo con el mismo nombre que la evidencia y extensión *p7s*, que es un "PKCS#7 Signature" (*Public-Key Cryptography Standard*), de acuerdo con la sección 3.2 del "RFC 2311" [24]. La huella digital se almacena en la base de datos de la *TSA* y devuelve al emisor el archivo de extensión *p7s*. La *TSA* vía herramienta alojada en su página web, ofrece la posibilidad de comprobar en el futuro la fecha y la hora de certificación del archivo (a modo de herramienta de auditoría).

Como cada vez que se envía a la *TSA* una solicitud de sello de tiempo se genera un archivo de extensión *p7s*, es posible el análisis forense de la CdC mediante el estudio de la correlación temporal de archivos.

Análisis de funcionalidad del paquete de evidencia. Se trata de demostrar que mediante un teléfono móvil inteligente o *Smartphone* (o Tableta, *Smartcamera*, etc), es posible obtener evidencias digitales, así como definir e iniciar una CdC.

Mediante el análisis de los metadatos asociados con la evidencia (datos contenidos en el archivo de imagen intercambiable, *Exif*), es posible analizar con más detalle las características de la prueba:

- Título de la prueba. Es conveniente no modificar el nombre que de forma automática genera el *Smartphone*, ya que incluye la fecha y hora de la adquisición de la prueba.
- Tipo de archivo de la prueba. Permite identificar si se trata de un archivo de audio, vídeo o imagen fotográfica.
- Fecha y hora de la captura de la evidencia.
- Carpeta donde la evidencia se guarda en el *Smartphone*.
- Nombre del lugar en donde se obtuvo la evidencia. Se basa en el sistema de geoposicionamiento *Google*, por tanto, es esencial que la opción esté habilitada en el sistema operativo y *3G/4G* o cobertura *WiFi WPA2 PSK* con clave robusta no contenida en diccionario.
- La geolocalización de la prueba (latitud y longitud), basada en el dispositivo *GPS* y/o servicio de *Google*.

- Tamaño de la evidencia, válido para indicar el camino a seguir en el tratamiento y la mejor manera de enviar la información.

- Resolución del archivo de imagen. Muestra la calidad de la información de las pruebas.

- La localización del archivo en la estructura de ficheros de la memoria del *Smartphone*: datos necesarios con el fin de tratar el archivo denominado "paquete de evidencia".

GPS	
Referencia de latitud GPS	Latitud norte
Latitud GPS	28.124170' 0"
Referencia de longitud GPS	Longitud oeste
Longitud GPS	15.424470' 0"
Referencia de altitud GPS	Nivel del mar
Altitud GPS	34.5 m
Marca de fecha y hora GPS	15:13:43
Método de procesamiento GPS	(41,53,43,49,49,00,0...
Marca de fecha GPS	2013:05:05
Misceláneo	
Versión Exif	2.2
Nota del fabricante	(05,00,01,00,07,00,0...
Versión de FlashPix	1.0
ID Versión GPS	(2,2,0,0)

Fig. 3. Detalle del fichero *Exif* del GPS.

La mayor parte de los datos incluidos en los metadatos se pueden utilizar al generar la información en el documento de pruebas y de control de cambios. La Figura 3 muestra los detalles del fichero *Exif* generado en la utilización del *GPS*.

Lo mismo sucede con la posición exacta para localizar el punto de adquisición de las pruebas. La mejor manera de confirmar dicho lugar es mediante el uso de *GPS*, pero tiene el inconveniente de sólo ser posible si el satélite es visible, ya que si no la información será aproximada.

Una vez que se ha obtenido la prueba y se han extraído los datos de identificación, es posible cifrar la evidencia. Para ello se puede recurrir al uso de aplicaciones de cifrado *AES* de al menos 128 bits.

Con la evidencia cifrada, será necesario enviar de manera segura el fichero a una *TSA* que proporcione un servicio de notaría electrónica. Si se utilizan los servicios de *DigiStamp*, se obtiene a nivel local una huella del tipo SHA-256 o SHA-512 bits. En cualquier momento se podrá verificar que el sello de tiempo ha sido generado por la *TSA*, así como el momento de generación.

Hay que señalar tres desventajas detectadas:

- La certificación es sólo para el momento en que se envía el archivo a la *TSA*, pero no indica la hora de la generación de evidencia.

- El trabajo de campo en el sitio web *DigiStamp* es imposible, ya que no está diseñado para funcionar en dispositivos móviles y no funciona con cualquier navegador (*Android, Opera, Firefox, Chrome*, etc.). Por lo tanto, es necesario transferir la información a un ordenador personal y utilizar un navegador de Internet.

- El servicio tiene un costo por fichero. Por dicho motivo, se podría crear algún sistema que funcionara directamente para nuestro propósito.

Documento de pruebas y control de cambios. Existen varias opciones para el formato del mismo (texto plano, *xml, doc*, etc.) La propuesta de este trabajo, por su sencillez y universalidad, es de texto plano. El documento deberá contener, como mínimo:

- Nombre detallado de la persona que adquiere la evidencia, la posición, la razón, el lugar, la hora y la fecha, las autorizaciones, el nombre de las evidencias y los nombres de los ficheros de sellado de tiempo (archivos *p7s*).

- Nombre detallado, la posición, la razón, la ubicación, la hora y fecha de cada persona a la que se envía el documento en la CdC.

- La certificación en clave asimétrica del documento completo, con indicaciones de principio y fin.

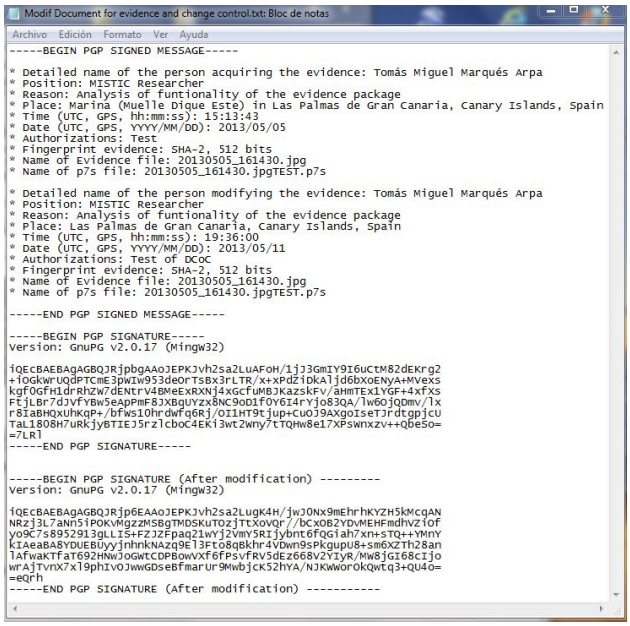


Fig. 4. Documento de pruebas y de control de cambios.

La Figura 4 es un ejemplo del documento, que ha sido firmado con una clave *RSA* asimétrica de 2048 bits, utilizando el programa *GnuPG* versión v2.0.1 para Windows 7. La aplicación del programa será necesaria cada vez que el documento avance en la CdC y se hagan modificaciones a firmar.

Propuesta de una aplicación en *Android*. Consiste en la creación de una aplicación para teléfonos móviles inteligentes. Debe ser capaz de capturar la evidencia, crear el paquete de evidencia y realizar envíos de correo electrónico o a un servidor seguro *SFTP* o *FTPS*.

La aplicación ha de tener en cuenta los componentes del equipo que necesiten ser activados. Una vez que la evidencia ha sido capturada, la aplicación será capaz de cifrar, realizar una conexión segura a una *TSA* y obtener los archivos *p7s*. Con la ayuda de los metadatos, será capaz de generar el documento de pruebas y de control de cambios, que estará firmado utilizando una clave asimétrica *RSA* de 2048 bits.

Al final ha de ser capaz de generar un archivo comprimido de la evidencia, formado por la propia evidencia, el archivo

p7s y el documento de pruebas y de control de cambios. Este fichero es el paquete de evidencia. Finálmente, el resultado deberá estar listo para ser enviado por email o preferentemente a un servidor seguro *SFTP*.

IV. Conclusiones y trabajos futuros

Este trabajo ha sido desarrollado con la intención de crear un método válido de CdC. En un principio la idea era sólo crear la cadena, pero con posterioridad se comprobó que ésta debía tener un punto de partida: la generación de la evidencia. Y fue allí donde se ha hallado lo que posiblemente sea el punto más débil de ella.

Por lo tanto, ¿se debe seguir un guion en la adquisición de pruebas para asegurar que se procede de manera correcta? La respuesta es que no. La tecnología actual puede permitir la automatización de ciertas tareas y rutinas, que es la propuesta principal de este trabajo mediante la creación de una herramienta que automatice el proceso en la parte más débil de la cadena: la correcta adquisición de la evidencia. Posteriormente, la prueba debe ser protegida de las mayores amenazas que se han detectado: *spoofing*, *jamming*, *phishing*, *man-in-the-middle*, *wiretapping*, colisión y preimagen. Para evitar esto y sobre todo, para que no se pueda modificar fácilmente la evidencia sin dejar rastros, se ha propuesto un método de trabajo.

Las dos cuestiones planteadas en las preguntas y objetivos de investigación: ¿es posible desarrollar un nuevo método para la Cadena de Custodia? y, ¿el nuevo método puede ser implementado? La respuesta es afirmativa en ambos casos, como se ha demostrado.

Principalmente los trabajos de mejora se pueden centrar en los siguientes aspectos:

- Creación de una aplicación *Android* en la forma propuesta.
- Diseño de un dispositivo hecho en una plataforma del tipo *"Raspberry Pi"* o *"BeagleBone Black"* (pequeños ordenadores de muy bajo coste que admiten conexión de periféricos) y que pueden ofrecer otras posibilidades en la creación de las CdC mediante la utilización de imágenes en lugar de huellas, y que por tanto eviten la recusación de una evidencia por la degeneración del soporte físico que la contiene.
- Uso de la identificación biométrica de los usuarios de acuerdo con el progreso técnico.
- Utilización de geolocalización lo más precisa y segura posible con la incorporación de datos de posicionamiento cifrados GNSS-PRS
- Uso de datos de la tarjeta *SIM* para proporcionar la identificación del usuario.
- Realización de ciberataques a la propuesta con el fin de demostrar su debilidad o su fortaleza.

Agradecimientos

This work was partly funded by the Spanish Government through projects: TIN2011-27076-C03-02 "CO-PRIVACY" and CONSOLIDER INGENIO 2010 CSD2007-0004 "ARES".

Referencias

[1] European Commission C-4536. "Líneas de investigación de la Comisión Europea para 2013". Cooperación, tema 10, seguridad. Área material: 10.1.4. Delincuencia común y forense - Topic SEC-2013.1.4-2.- *Desarrollo de un Marco Común Europeo para la aplicación de las nuevas tecnologías en la recopilación y el uso de la evidencia*, julio 2012.

[2] V. Vaishnavi, W. Kuechler. "Design research in information systems", 2004 (revisión octubre 2013). Disponible en http://desrist.org/design-research-in-information-systems/.

[3] B. J. Oates. "Researching information Systems and Computing". SAGE Publications Ltd. London, 2006, (revisión 2013).

[4] T. F. Gayed, H. Lounis, M. Bari. "Cyber Forensics: Representing and (Im)Proving the Chain of Custody Using the Semantic web". *COGNITIVE 2012: The Fourth International Conference on Advanced Cognitive Technologies and Applications*, 2012.

[5] G. Giova. "Improving Chain of Custody in Forensic Investigation of Electronic Digital Systems". *International Journal of Computer Science and Network Security*, vol. 11, no. 1, 2011.

[6] S. L. Garfinkel. "Providing cryptographic security and evidential chain of custody with the advanced forensic format, library and tools". Naval Postgraduate School & Harvard University, USA, 2011.

[7] J. Ćosić, M. Bača. "A Framework to (Im)Prove "Chain of Custody" in Digital Investigation Process". *Proceedings of the 21st Central European Conference on Information and Intelligent Systems*, 2010.

[8] J. Ćosić, M. Bača. "(Im)Proving Chain of Custody and Digital Evidence Integrity with Time Stamp". Universidad de Zagreb, 2010.

[9] M. Colobrán. "Análisis Forense de la Información". *Conceptos básicos*. MISTIC. Universitat Oberta de Catalunya, 2012.

[10] M.R. Clint, M. Reith, G. Gunsch. "An Examination of Digital Forensic Models", 2002.

[11] B.D. Carrier. "Defining Digital Forensic Examination and Analysis Tools". *International Journal of Digital Evidence*, 2002.

[12] P. G. Bradford, D. A. Ray. "An Online Algorithm for Generating Fractal Hash Chains Applied to Digital Chains of Custody". *Intelligence and Security Informatics 2007 Conference (ISI 2007)*.

[13] J. A. Colquitt. "Alabama Law of Evidence". The Mitchie Company–Law Publishers, Charlottesville, VA, 1990.

[14] National Institute of Justice, USA. "Crimes Scene Guides", 2011 (acceso febrero 2014). Disponible en http://www.ojp.usdoj.gov/nij/topics/law-enforcement/investigations/crime-scene/guides/glossary.htm.

[15] A. Guash. "Análisis Forense de la Información". *El informe pericial. Análisis forense y sistema legal*. MISTIC. Universitat Oberta de Catalunya, 2012.

[16] J. Tallim. "Deconstructing Web Pages". *Media Smarts*, 2012 (accesso febrero 2014). Disponible en http://mediasmarts.ca/.../deconstructing-web-pages-lesson.

[17] B. D. Carrier, E.H. Spafford. "An Event-Based Digital Forensic Investigation Framework". *DFRWS*, 2004.

[18] C. Willassen. "Hypothesis based investigation of Digital Time Stamp". *FIP, Advanced in Digital Forensic IV*, pp.75–86, 2008.

[19] S. Vanstone, P. van Oorschot, A. Menezes. "Handbook of Applied Criptografy". CRC Press, 1997.

[20] C. Strawn "Expanding the Potential for GPS". *Evidence Acquisition, Small Scale digital evidence Forensic Journal*, vol.3, no.1, 2009.

[21] J. L. García Rambla. "Un forense llevado a juicio". *Prueba anticipada en un proceso civil*, cap.7. Flu-Proyect y Sidertia Solutions, *Creative Commons*, 2013.

[22] NIST 197. "Advanced Encryption Standard (AES)". *Federal Information Processing Standards. Special Publication 197*. National Institute of Standards and Technology (NIST), Maryland, USA, 2001.

[23] Blue Book. "Recommendation for Space Data System Standards". *CCSDS Cryptographic Algorithms Recommended Standard CCSDS 352.0-B-1*. CCSDS Secretariat. Space Communications and Navigation Office. NASA Headquarters, Washington, USA, 2012.

[24] S. Dusse, USSE, P. Hoffman, B. Ramsdell, L. Lundblade, L. Repka "RFC 2311". *S/MIME Version 2 Message Specification*. ISOC, Virginia, USA, 1998.

Esteganografía en zonas ruidosas de la imagen

Daniel Lerch-Hostalot
Universitat Oberta de Catalunya,
Internet Interdisciplinary Institute (IN3),
Estudis d'Informàtica, Multimèdia i Telecomunicació,
Rambla del Poblenou, 156,
08018 Barcelona,
Email: dlerch@uoc.edu

David Megías
Universitat Oberta de Catalunya,
Internet Interdisciplinary Institute (IN3),
Estudis d'Informàtica, Multimèdia i Telecomunicació,
Rambla del Poblenou, 156,
08018 Barcelona,
Email: dmegias@uoc.edu

***Resumen*—La mayor parte del estegoanálisis en el estado del arte se basa en el uso de técnicas de *machine learning*, es decir, en entrenar clasificadores para que sean capaces de diferenciar una imagen portadora de una imagen con mensaje oculto. Las investigaciones realizadas en este campo muestran que las zonas de la imagen más difíciles de modelar y, en consecuencia, aquellas en las cuales es más difícil detectar un mensaje incrustado, son las zonas ruidosas. Estas corresponden a líneas y texturas. En este artículo presentamos un nuevo método de esteganografía que permite ocultar información en dichas zonas, dificultando así su detección. La efectividad del método se ha comprobado usando dos bases de datos de imágenes diferentes y dos estegoanalizadores recientes. Los experimentos demuestran que el algoritmo propuesto mejora significativamente la indetectabilidad estadística respecto al sistema LSB *matching* para la misma capacidad de incrustación.**

***Palabras clave*—Esteganografía, Estegoanálisis.**

I. Introducción

La esteganografía estudia diferentes técnicas para la ocultación de datos en otros objetos, conocidos como objetos portadores. Actualmente, estos objetos portadores suelen ser medios digitales, como por ejemplo imágenes, vídeos o archivos de sonido. No obstante, sin lugar a dudas, el medio más utilizado en la actualizad son las imágenes, por su amplia difusión en Internet.

Uno de los métodos más usados para ocultar información en imágenes de mapas de bits es la sustitución del bit menos significativo (*Least Significant Bit*, LSB). Este método divide el mensaje original en bits y oculta cada uno de ellos en un píxel de la imagen. La variación en el valor del píxel es tan poco significativa que no puede ser detectada visualmente, pero resulta suficiente para ocultar información. Sin embargo, como puede verse en [16], esta técnica presenta algunos inconvenientes. La sustitución del LSB es una operación asimétrica, pues los píxeles con un valor par tenderán a incrementar su valor (cuando se incruste un '1'), mientras que los píxeles con un valor impar tenderán a disminuirla (cuando se incruste un '0'). Esto crea anomalías estadísticas en la imagen, como por ejemplo parejas de barras (frecuencias) que tienden a igualarse en el histograma de luminosidad de la imagen [16]. Finalmente, esta debilidad de la sustitución del LSB ha culminado en ataques como el RS [5] o el SPA [4], los cuales han conseguido detectar la presencia de información oculta incluso cuando el número de bits incrustados apenas alcanza el 3 % del número de píxeles de la imagen. Debido a estos ataques, el método de sustitución del LSB ha dejado de ser considerado como seguro.

El sistema que ha tomado el relevo de la sustitución del LSB es el conocido como LSB *matching* [15]. Este método es muy similar al anterior, pues solo tiene una pequeña diferencia: en lugar de sustituir el valor del LSB directamente por el bit a incrustar, lo que hace es modificarlo sumando o restando uno al valor total del píxel cuando el bit a incrustar no coincide con el LSB del píxel correspondiente. El efecto sobre el LSB es el mismo, así como la dificultad para detectarlo visualmente. Sin embargo, al proceder de esta manera, ya no se trata de una operación asimétrica y no se introducen anomalías estadísticas tan evidentes. De hecho, con este método resulta muy difícil diferenciar un mensaje oculto del ruido existente en todas las imágenes que aparece como consecuencia del proceso de captura.

Para detectar este método de ocultación de información en imágenes los estegoanalistas han recurrido al uso de técnicas de *machine learning* [2]. Para ello es necesario preparar una base de datos de imágenes que se usarán para entrenar un clasificador y verificar que este funciona correctamente. Este clasificador será el encargado de diferenciar las imágenes con mensaje oculto (imágenes esteganográficas) de las imágenes no alteradas (imágenes portadoras). En este tipo de estegoanálisis el trabajo del estegoanalista se basa principalmente en detectar aquellas características de la imagen que son más susceptibles de ser alteradas cuando se oculta información. Estas características son las usadas para entrenar al clasificador. Si bien se han propuesto diferentes métodos de estegoanálisis basados en clasificadores que ofrecen buenos resultados [13], [8], [11], todavía queda mucha investigación para avanzar en este campo.

Una de las lecciones aprendidas en los últimos años de investigación en estegoanálisis usando clasificadores es que existen zonas que son mucho más difíciles de modelar que otras: los bordes y las texturas. Estas zonas contienen mucho ruido y, en ellas, es muy difícil extraer características ade-

Este trabajo está financiado parcialmente por el Ministerio de Economía y Competitividad a través de los proyectos TIN2011-27076-C03-01/02 "CO-PRIVACY" y CONSOLIDER INGENIO 2010 CSD2007-0004 "ARES".

cuadas para entrenar al clasificador. Esta es la base que usan algunas técnicas modernas de esteganografía, como por jemplo las presentadas en [14], [9]. En este artículo se presenta una nueva técnica de ocultación de información que usa estas zonas ruidosas de la imagen para incrustar los bits del mensaje.

El resto del artículo se organiza de la manera siguiente. En la Sección II se presenta el método de esteganografía propuesto en el artículo. En la Sección III se analiza experimentalmente el método y se comprueba su indetectabilidad usando software de estegoanálisis. Finalmente, la Sección IV presenta las conclusiones extraídas de este trabajo.

II. Método propuesto

II-A. Motivación

Los métodos modernos de estegoanálisis usan clasificadores para modelar las propiedades estadísticas de las imágenes, pero existen zonas que son especialmente difíciles de modelar, como los bordes o las texturas. Por ello, la investigación en nuevos sistemas de esteganografía se centra, en gran parte, en la construcción de métodos que permitan ocultar información en esas zonas.

Sistemas de estegoanálisis como [13], [8], [11] modelan las características extraídas de la imagen como diferencias entre píxeles vecinos. Tomemos como ejemplo el método basado en patrones de diferencias de píxeles (*patterns of pixel diferences*, PPD) presentado en [11]. En ese artículo se usan cinco píxeles vecinos para modelar la imagen, como se muestra en la Fig.1, tomando uno de ellos como referencia a restar de los demás.

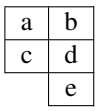


Figura 1: Extracción de características basadas en bloques PPD

De esta forma pueden obtenerse vectores formados por cuatro posibles diferencias. Por ejemplo, si tomamos b como base, podemos obtener una representación en cuatro dimensiones: $v = [a-b, c-b, d-b, e-b]^{\mathrm{t}}$, donde $[\cdot]^{\mathrm{t}}$ denota el operador de transposición de un vector (o una matriz).

Suponiendo el caso más sencillo, es decir, usando imágenes en escala de grises con una profundidad de color de 8 bits, cada píxel puede tomar un valor de 0 a 255. Por lo tanto, en el peor de los casos, la diferencia entre dos píxeles vecinos será de 255. Así pues, con el modelo presentado, podrían generarse hasta 255^4 características, lo que resulta numéricamente impracticable para los clasificadores actuales. Con los sistemas propuestos en [13] y [8] la situación es similar. No obstante, no es habitual que un píxel con valor 0 sea el vecino de un pixel con valor 255, dado que en las imágenes el valor de los píxeles suele cambiar en forma de degradado. Por lo tanto, ignorar diferencias muy grandes entre valores vecinos no suele perjudicar a los sistemas de estegoanálisis. Es por ello que se utiliza un parámetro para reducir el número de características. Por ejemplo, en [11] se sugiere el uso de un umbral $S = 4$. De esta manera, en lugar de obtener 255^4 características, se obtienen 4^4, que es una cifra mucho más manejable. Con esta aproximación lo métodos de estegoanálisis pueden atacar el problema sin que la explosión en el número de características les impida modelar la imagen.

Sin embargo, este no es el único motivo para el uso de un umbral para reducir el número de dimensiones. Otro problema asociado a la dimensionalidad, y que perjudica seriamente al estegoanálisis, es el de la obtención de muestras insuficientes. Este tipo de problema, detectado previamente en estegoanálisis [7], se produce al usar modelos de grandes dimensiones. Cuantas más dimensiones tenga el modelo, más difícil es encontrar muestras en la imagen para todas las posibilidades que ofrece. Durante la extracción de características, se reparten todas las muestras extraídas de la imagen entre cada uno de los patrones de los que dispone el modelo. En el caso de PPD, por ejemplo, existen T^4 patrones, la frecuencia de los cuales dependerá de la imagen y de su contenido. Dado que el número total de muestras es fijo y a medida que crece el valor de T aumenta el número de patrones, la frecuencia de cada patrón será cada vez más pequeña. Los patrones menos frecuentes serán los primeros en llegar a frecuencias tan bajas que su valor no será representativo y perjudicarán al entrenamiento del clasificador. Por este motivo existe un límite en el valor del umbral a partir del cual los métodos de estegoanálisis dejan de ser efectivos. El método que se propone en este artículo, pretende explotar esta debilidad.

II-B. Zonas de inserción

Como se ha describe en el apartado anterior, el objetivo del método presentado en este artículo es ocultar la información en las zonas más difíciles de modelar de la imagen. El problema que se presenta cuando se desea ocultar información únicamente en unas zonas concretas de la imagen es cómo comunicar al receptor del mensaje (de la imagen) en qué zonas debe leer y en qué zonas no. Si se desarrolla un procedimiento para identificar las zonas ruidosas, estas pueden cambiar (dejar de ser ruidosas) al ocultar información, por lo que el receptor puede acabar leyendo en zonas donde no hay mensaje e ignorando zonas donde sí lo había.

Un enfoque válido, tomado en los métodos [14], [9], es el uso de *Wet Paper Codes* (WPC) [6]. Estos métodos, que son muy adecuados para el problema presentado, son relativamente complejos, lo que implica un procesamiento muy lento en la inserción del mensaje. En este trabajo se expone un procedimiento alternativo, que resulta muy rápido y sencillo en comparación con el uso de WPC.

Para detectar las zonas de inserción se establecerá un umbral T que nos indicará las zonas difíciles de modelar, de la misma forma que lo hacen los sistemas de estegoanálisis. Se agruparan los píxeles en parejas de píxeles vecinos (a, b),de manera que solo los tomaremos en consideración para ocultar información si su diferencia es mayor o igual al umbral T, es decir si $|a-b| \geq T$.

(a) Imagen Lenna en escala de grises (b) Zonas de ocultación para $T = 4$ (c) Zonas de ocultación para $T = 10$

Figura 2: Imagen Lenna y zonas donde se oculta la información para diferentes valores de T

En la Fig.2(b) se muestran, con píxeles negros, las zonas donde se ocultaría la información en caso de usar $T = 4$ para la imagen Lenna de la Fig.2(a). Análogamente, la Fig.2(c) muestra las zonas de ocultación de información para el caso $T = 10$. En ellas podemos ver cómo se evitan las zonas más uniformes de la imagen, mientras que los bordes, principalmente, y algunas texturas son las zonas que se usan para ocultar información.

Para ocultar información el procedimiento consiste en recorrer la imagen tomando cada vez una pareja diferente de píxeles vecinos. Las parejas se forman sin solapamiento, de manera que, dados cuatro píxeles vecinos, (a, b, c, d), se formarán las parejas (a, b) y (c, d), mientras que la pareja (b, c) no se tendrá en consideración. Las parejas así formadas no se usarán para incrustar información si su diferencia esta por debajo de umbral T. La imagen se puede recorrer tomando las parejas de forma horizontal o vertical, aunque esto no es determinante para el funcionamiento del método propuesto.

II-C. Procedimiento de incrustación

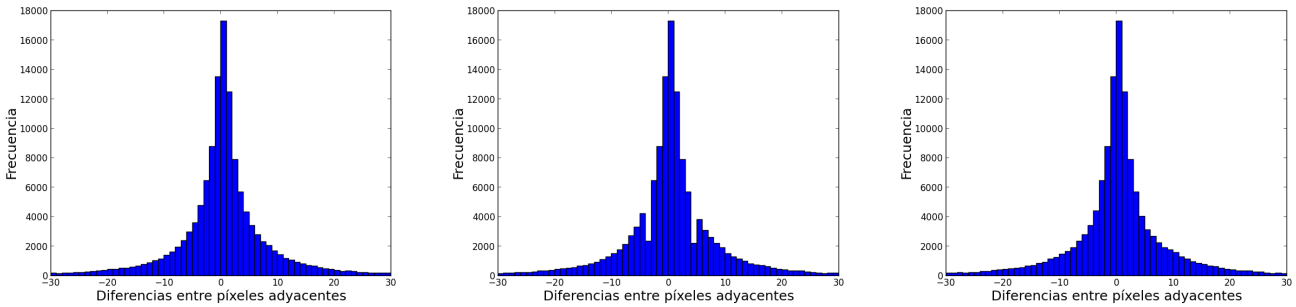
El método propuesto usa cada pareja que supera el umbral para ocultar un bit. Concretamente se oculta alterando el píxel de la izquierda, es decir, el etiquetado como a de la pareja (a, b). Para ello, se usa el LSB de a como bit de información, dejándolo tal y como está si su valor es igual al del bit del mensaje que se quiere ocultar y modificándolo si su valor no coincide. Esta modificación se realizará siempre incrementando la diferencia entre los valores de los píxeles de la pareja. Así pues, a', el nuevo valor de a, vendrá determinado por la ecuación siguiente:

$$a' = \begin{cases} a, & \text{si } a \bmod 2 = m, \\ a+1, & \text{si } a > b \\ a-1, & \text{si } a < b. \end{cases}$$

La idea es incrementar siempre la diferencia entre los píxeles de la pareja, nunca disminuirla. El motivo es que incrementar o disminuir aleatoriamente, de forma similar a como se hace en LSB *matching*, llevaría en el caso $|a - b| = T$ a dejar de cumplir el umbral establecido para algunas parejas, que pasarían a tener una diferencia $T - 1$. Al no cumplir el umbral de lo que consideramos un píxel adecuado para ocultar información, el receptor no sabría que debe leer información de él y se perdería la información oculta en ese píxel.

Lógicamente, esta operación para ocultar información introduce una anomalía estadística, pues no se aplica de forma equitativa sobre todas las parejas de píxeles. Al incrementar la diferencia entre parejas con un valor superior a $T+1$, parte de esas parejas (en las que se quiere ocultar un bit diferente al LSB de a) se convierten en parejas con diferencia $T + 2$. Análogamente, algunas parejas con diferencia $T + 2$ pasan a tener diferencia $T+3$ y así sucesivamente. En general, aunque varias parejas con diferencia $T + n$ pasan a tener diferencia $T + n + 1$, este hecho se ve compensado por el número de parejas nuevas con diferencia $T+n$ que aparecen al incrustar información en parejas con diferencia $T + n - 1$. Pero hay un caso especial, el de las parejas con diferencia T, pues mientras que parte de estas pasan a tener diferencia $T + 1$, el número de parejas con diferencia T no se ve retroalimentado y solo decrece. Esta anomalía puede observarse en el histograma de los valores de las diferencias entre píxeles adyacentes representado en la Fig.3(b), en comparación con el histograma de la imagen original que aparece en la Fig.3(a).

Para eliminar esta anomalía en el histograma, se puede repartir la responsabilidad de ser la primera pareja (la que tiene diferencia igual a T) entre diferentes parejas. De esta forma, se consigue que las barras del histograma correspondientes no decrezcan lo suficiente como para generar una anomalía. Para ello, se usa un valor de T dinámico, que dependerá del pixel que se esté modificando. La idea es inicializar un generador de números pseudoaleatorios (*Pseudo-Random Number Generator*, PRNG) con una semilla (por ejemplo una contraseña) que deberán conocer tanto el emisor del mensaje como el receptor. Este PRNG se utiliza para generar una secuencia de valores dinámicos para el umbral T. Se usa un rango de



(a) Imagen original

(b) Imagen esteganográfica con $T = 4$

(c) Imagen esteganográfica con $T_{\text{mín}} = 4$ y $T_{\text{máx}} = 9$

Figura 3: Histograma de diferencias entre parejas de píxeles adyacentes para la imagen original, la imagen esteganográfica con $T = 4$ y la imagen esteganográfica con umbral dinámico entre $T_{\text{mín}} = 4$ y $T_{\text{máx}} = 9$

valores de T entre un mínimo, $T_{\text{mín}}$, y un máximo, $T_{\text{máx}}$. De esta manera los valores dinámicos de T se generarán con la función $T = \text{PRNG}(T_{\text{mín}}, T_{\text{máx}})$, que devuelve un número pseudoaleatorio con distribución uniforme en el intervalo $[T_{\text{mín}}, T_{\text{máx}}]$. De esta manera, como se puede apreciar en la Fig.3(c), la anomalía en el histograma desaparece.

El uso del PRNG para seleccionar un valor dinámico del umbral no solo sirve para eliminar la anomalía estadística, sino que además ofrece una capa de seguridad adicional que dificulta el estegoanálisis, dado que no se puede saber con exactitud en qué píxeles se ha ocultado información sin disponer de la semilla.

II-D. Algoritmos de incrustación y de extracción

Tal y como se explica en los apartados anteriores, ya tenemos todas las piezas necesarias para el algoritmo de inserción de datos de la imagen (Algoritmo 1).

La extracción de datos es similar. Basta con inicializar el PRNG, recorrer la imagen extrayendo parejas de píxeles e ir leyendo los LSB del primer píxel de cada pareja que cumple con el umbral T (Algoritmo 2).

III. Resultados experimentales

El método se ha verificado con dos sistemas de estegoanálisis: PPD [11] y SPAM [13] (en la versión más efectiva de este, que usa características de segundo orden). Estos sistemas permiten extraer características de las imágenes. Como clasificador, se ha usado una implementación de una *Support Vector Machine* (SVM) [3], por ser uno de los clasificadores que ofrece mejores resultados en estegoanálisis.

La SVM debe ser ajustada para que proporcione unos resultados óptimos. Concretamente, es necesario seleccionar valores para los parámetros C y γ. Estos valores serán escogidos para dar al clasificador la capacidad de generalizar. Para escoger dichos parámetros, se ha seguido el proceso especificado en [10], es decir, realizando una validación cruzada en el conjunto de entrenamiento de todos los posibles valores de los parámetros C y γ que se especifican a continuación:

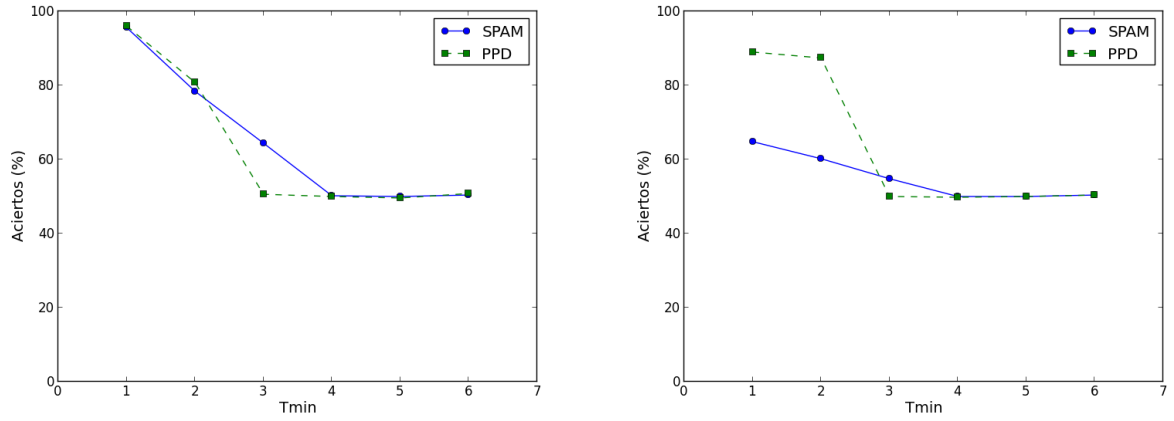
Algorithm 1 Ocultar mensaje

Input: $M, I, \text{Seed}, T_{\text{mín}}, T_{\text{máx}}$
M: Mensaje a ocultar
I: Matriz $[1..H, 1..W]$ que contiene la imagen original
Seed: Semilla del generador PRNG
$T_{\text{mín}}$: Valor mínimo para el cálculo de T
$T_{\text{máx}}$: Valor máximo para el cálculo de T
Output: I'
I': Matriz que contiene la imagen con el mensaje oculto

1: InitializePRNG(Seed)
2: **for all** $i \in [1, H]$ **do**
3: **for all** odd $j \in [1, W-1]$ **do**
4: $T \leftarrow \text{PRNG}(T_{\text{mín}}, T_{\text{máx}})$
5: $a \leftarrow I[i, j]$
6: $b \leftarrow I[i, j+1]$
7: **if** $|a - b| \geq T$ **then**
8: $m \leftarrow \text{nextBit}(M)$
9: $I'[i,j] \leftarrow \begin{cases} a, & \text{if } a \bmod 2 = m, \\ a+1, & \text{if } a > b, \\ a-1, & \text{if } a < b. \end{cases}$
10: **end if**
11: **end for**
12: **end for**

$$C \in \left\{2^{-5}, 2^{-3}, 2^{-1}, 2^{1}, 2^{3}, \ldots, 2^{15}\right\},$$
$$\gamma \in \left\{2^{-15}, 2^{-13}, 2^{-11}, \ldots, 2^{-1}, 2^{1}, 2^{3}\right\}.$$

Los experimentos se han realizado con la base de datos BOSS, presentada en [1], por ser una de las más usadas en esteganografía, y también en la base de datos pública NRCS [12], por disponer de imágenes de alta resolución muy ruidosas, significativamente diferentes de las de BOSS. Para cada base de datos, se han creado dos grupos de imágenes, uno que se usa como conjunto de entrenamiento y otro que se usa como conjunto de verificación. Cada uno de ellos esta formado por 500 imágenes, 250 de ellas sin incrustar (portadoras) y



(a) Base de datos BOSS

(b) Base de datos NRCS

Figura 4: Porcentajes de detección correcta en función de $T_{\text{mín}}$, usando las bases de datos de imágenes BOSS y NRCS

Algorithm 2 Extraer mensaje

Input: I, Seed, $T_{\text{mín}}, T_{\text{máx}}$
I: Matriz $[1..H, 1..W]$ que contiene la imagen con el mensaje oculto
Seed: Semilla del generador PRNG
Output: M
M: Mensaje extraído

```
InitializePRNG(Seed)
for all i ∈ [1, H] do
    for all odd j ∈ [1, W − 1] do
        T ← PRNG(T_mín, T_máx)
        a ← I[i, j]
        b ← I[i, j + 1]
        if |a − b| ≥ T then
            M ← addBit(M, a)
        end if
    end for
end for
```

las otras 250 con información incrustada (imágenes esteganográficas). El umbral usado por defecto en PPD es $T = 4$ (parámetro S especificado en [11], mientras que en SPAM es de $T = 3$. Los experimentos se han diseñado para verificar que, marcando con umbrales superiores a los establecidos por las herramientas de estegoanálisis, el método presentado no se detecta. Se ha incrustado información en las imágenes usando diferentes valores para $T_{\text{mín}}$ y $T_{\text{máx}}$, tal y como se muestra en el Cuadro I.

Como se puede ver en la Fig.4, el porcentaje de detección cae al 50 % (equivalente a decisión aleatoria, o sea, a no detección) aproximadamente al llegar a $T_{\text{mín}} = 4$. En los gráficos se aprecia como los métodos de estegoanálisis fallan cuando la información esta oculta en zonas que no pueden modelar. Los experimentos se han realizado sobre dos bases de datos de imágenes muy diferentes, y en ambos casos, el algoritmo propuesto no es detectado cuando $T_{\text{mín}}$ supera el umbral usado por los métodos de estegoanálisis.

Cuadro I: Valores mínimos y máximos del umbral usados en los experimentos

$T_{\text{mín}}$	$T_{\text{máx}}$
1	6
2	7
3	8
4	9
5	10
6	11

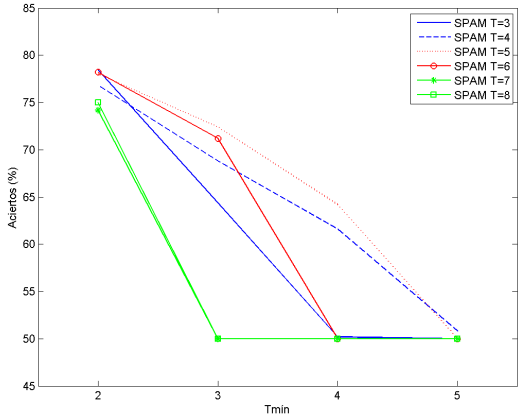


Figura 5: Porcentaje de detección correcta en función de $T_{\text{mín}}$, usando la base de datos de imágenes BOSS y diferentes valores de T, para el método de estegoanálisis SPAM

Sin embargo, podría parecer que el motivo por el que los métodos de estegoanálisis no detectan al método de esteganografía propuesto es por la elección de un umbral superior

al que ellos usan y que bastaría con subir también el umbral usado en estos métodos. Pero esto no es así dado que, si se incrementa el umbral, aumenta el número de dimensiones y aparecen combinaciones para las que no existen muestras o para los que existen muy pocas. Esto, como se ha comenta en la Sección II, empeora considerablemente los resultados del estegoanálisis. En la Fig.5 se puede observar que no existe ningún umbral T que permita detectar el método propuesto con SPAM si $T_{\text{mín}} \geq 5$. Lo mismo sucede para PPD si $T_{\text{mín}} \geq 4$.

La cantidad de información que puede incrustarse (es decir, la capacidad del método) en una imagen depende de las zonas ruidosas de esta, por lo que no es sencillo de determinar *a priori*. A nivel orientativo, usando $T_{\text{mín}} = 4$ y $T_{\text{máx}} = 9$ en las imágenes de BOSS, se ha realizado la inserción con una ratio media del 9 %, es decir incrustando un bit en el 9 % de los píxeles (0,09 bits por píxel). En las imágenes de NRCS la ratio de incrustación es del 13 %. Para mostrar la efectividad del sistema propuesto, se han comparado los resultados de indetectabilidad de este con los obtenidos usando la esteganografía LSB *matching* tradicional [15]. Para ello, se han usado los estegoanalizadores PPD y SPAM con el objetivo de calcular los porcentajes de detección cuando se incrusta en LSB *matching* usando una ratio del 9 % en BOSS y del 13 % en NRCS. De esta manera se puede realizar una comparación en igualdad de condiciones en cuanto a capacidad se refiere. Los resultados de detección se muestran en II. Como se puede observar, para las mismas ratios de inserción que no se detectan (porcentaje de aciertos del 50 %) con el algoritmo presentado, la esteganografía LSB *matching* se detecta con los estegoanalizadores SPAM y PPD (porcentaje de aciertos superior al 50 %).

Cuadro II: Detección de la esteganografía LSB *matching*, para la misma capacidad que el método propuesto, usando PPD y SPAM

Base de datos	Método de detección	Porcentaje de aciertos
BOSS	SPAM	85.00 %
BOSS	PPD	81.60 %
NRCS	SPAM	58.00 %
NRCS	PPD	64.00 %

IV. Conclusión

En este artículo se presenta un nuevo método para ocultar información en zonas difíciles de modelar de la imagen. Para detectar estas zonas, el método propuesto intenta explotar dos debilidades de los sistemas de estegoanálisis existentes: el crecimiento exponencial del número de características y la imposibilidad de extraer información útil de patrones con pocas muestras. Ambas debilidades tienen en común un umbral T, usado como base en el método presentado.

Por otra parte, se trata de un método que no requiere de ningún cálculo complejo, a diferencia de otros que persiguen objetivos similares, como los basados en WPC, por lo que es adecuado para entornos en los que la velocidad de ejecución o el rendimiento sean un factor clave. Los resultados muestran la indetectabilidad del método ante dos sistemas de estegoanálisis: PPD [11] y SPAM [13], viendo como la selección de un umbral T adecuado es suficiente para eludir la detección. También se comprueba que el ajuste del parámetro T en los métodos de estegoanálisis no permite la detección del método propuesto.

En futuros trabajos sería interesante estudiar si existen otros modelos similares que tengan en cuenta grupos de píxeles mayores que una pareja y si esto puede mejorar el algoritmo. Además, sería recomenable realizar un estudio teórico para el cálculo del valor óptimo de T.

Referencias

[1] T. Filler, T. Pevný, and P. Bas, *Break our steganographic system (BOSS)*, 2010. [Online]. Disponible: http://exile.felk.cvut.cz/boss/ [Accedido el 14 de julio de 2014].

[2] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics Series)*. New York, Secaucus, NJ, USA: Springer, 2006.

[3] C.-C. Chang and C.-J. Lin, "LIBSVM - A Library for Support Vector Machine," [Online]. Disponible: http://www.csie.ntu.edu.tw/~cjlin/libsvm [Accedido el 14 de julio de 2014].

[4] S. Domitrescu, X. Wu and N. D. Memon, "On Steganalysis of Random LSB Embedding in Continuous-tone Images," In *Proc. International Conference on Image Processing, ICIP 2002*, Rochester, NY, USA: IEEE, pp. 324-339.

[5] J. Fridrich, M. Goljan and R. Du, "Detecting LSB steganography in color and grayscale Images," In *Proc. ACM Workshop on Multimedia and Security*, Ottawa, Canada: ACM, pp. 22-28, 2001.

[6] J. Fridrich et al., "Writing on Wet Paper," *IEEE Trans. on Signal Processing*, vol. 53, no. 10, Oct. 2005, pp. 3923-3935.

[7] J. Fridrich, J. Kodovský, V. Holub, M. Goljan. "Breaking HUGO - the process discovery ," *Information Hiding, 13th International Workshop, Lecture Notes in Computer Science*.

[8] J. Fridrich and J. Kodovský, "Rich Models for Steganalysis of Digital Images," *IEEE Trans. on Information Forensics and Security*, vol. 7, no. 3, June 2012, pp. 868-882.

[9] V. Holub and J. Fridrich, "Designing Steganographic Distortion Using Directional Filters," In: *Proc. IEEE Workshop on Information Forensic and Security (WIFS)*, Tenerife, Spain: IEEE, pp. 234-239, 2012.

[10] C. W. Hsu, C. C. Chang, and C.J. Lin, "A practical guide to support vector classification," Department of Computer Science, National Taiwan University, 2003. [Online]. Disponible: http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf [Accedido el 14 de julio de 2014].

[11] D. Lerch-Hostalot and D. Megías, "LSB matching steganalysis based on patterns of pixel differences and random embedding," *Computers & Security*, vol. 32, Feb. 2013, pp. 192-206.

[12] National Resource Conservation Service, *NRCS Photo Gallery*, [Online]. Disponible: http://photogallery.nrcs.usda.gov [Accedido el 14 de julio de 2014].

[13] T. Pevný, P. Bas and J. Fridrich, "Steganalysis by Subtractive Pixel Adjacency Matrix," In *Proc. ACM Multimedia and Security Workshop*, Princeton, NJ, USA: ACM, pp. 75-84, 2009.

[14] T. Pevný, T. Filler, P. Bas, *Using High-Dimensional Image Models to Perform Highly Undetectable Steganography*. Information Hiding. Lecture Notes in Computer Science Volume 6387, 2010, pp 161-177.

[15] T. Sharp, "An Implementation of Key-Based Digital Signal Steganography," In *Information Hiding, Lecture Notes in Computer Science*, vol. 2137, Berlin-Heidelberg, Germany: Springer, 2001, pp. 13-26.

[16] A. Westfeld and A. Pfitzmann, "Attacks on Steganographic Systems," In *Information Hiding, Lecture Notes in Computer Science Volume*, vol. 1768, Berlin-Heidelberg, Germany: Springer, 2000, pp. 61-76

FastTriaje: un asistente para la clasificación de víctimas en situaciones de emergencia con autenticación robusta

Candelaria Hernández Goya
Departamento de
Ingeniería Informática
Universidad de La Laguna
Email: mchgoya@ull.edu.es

Alexandra Rivero García
Departamento de
Ingeniería Informática
Universidad de La Laguna
Email: alexandra.rivero.00@ull.edu.es

Pino Caballero Gil
Departamento de
Ingeniería Informática
Universidad de La Laguna
Email: pcaballe@ull.edu.es

***Resumen*—Este trabajo describe el desarrollo de un sistema de clasificación de víctimas en situaciones de emergencia que consta de una plataforma web y una aplicación móvil. La sinergia entre estos elementos y la integración de diferentes tecnologías de comunicación (NFC y Wi-Fi) permite clasificar a las posibles víctimas de una forma rápida y confiable.**

La clasificación realizada para una víctima es guiada por el dispositivo móvil (la implementación se ha hecho para teléfonos inteligentes basados en Android) y una vez finalizada, el sistema ofrece la posibilidad de almacenarla en una tarjeta NFC que se asigna a la víctima. Además, el diagnóstico realizado se almacena en el dispositivo móvil, pudiéndose posteriormente remitir a un servidor central en caso de que las infraestructuras de comunicaciones estén disponibles.

Se han implementado métodos criptográficos robustos, concretamente se utilizan Demostraciones de Conocimiento Nulo para identificar a los usuarios que desarrollan los triajes, de manera que sean sólo los usuarios autorizados previamente por el servicio los que puedan hacer uso del sistema.

***Palabras clave*—Triaje, NFC, Android, Criptografía Ligera, Demostración de Conocimiento Nulo (Zero Knowledge Proof)**

I. Introducción

El sistema presentado en este trabajo se denomina FastTriaje y se basa en el método START (Simple Triage and Rapid Treatment) [1]. Este método persigue dos metas esenciales en situaciones de emergencia y/o desastres naturales: salvar el mayor número de vidas posible y, simultáneamente, optimizar el uso de los recursos materiales y humanos disponibles.

Un sistema tradicional de triaje facilita la toma de decisiones sobre la prioridad requerida para la atención de una víctima por medio de tres acciones principales: observación, evaluación y decisión. Con FastTriaje el proceso completo de diagnóstico de la gravedad del paciente es guiado por la aplicación. El sistema indicará al diagnosticador el resultado final de la evaluación, así como la decisión que debe tomarse.

Uno de los elementos del sistema es una aplicación móvil desarrollada para dispositivos Android. Dicha aplicación será la herramienta principal del personal a cargo de la evaluación del paciente, tanto a la hora de realizar su clasificación, como también para gestionar la información generada en cada triaje. Una de las posibilidades incluida en la aplicación es almacenar la información asociada al triaje en una etiqueta NFC que se asociará al paciente. Dicha información puede ser consultada posteriormente en cualquier momento a través de la misma aplicación.

El segundo pilar del sistema desarrollado es una plataforma web cuya función principal es centralizar la recogida de información generada por el uso de la aplicación móvil y facilitar la gestión de la misma, incluyendo la gestión de usuarios y sus privilegios

La aplicación móvil y la plataforma web interactúan a través de un servicio web REST, siendo el formato seleccionado para la comunicación mensajes JSON.

Se han incluido métodos de autenticación robusta basados en criptografía ligera para la comunicación de la aplicación con la etiqueta NFC y también con la plataforma web.

II. Descripción de los sistemas de triaje

El termino triaje es de origen francés y su raíz (trier) significa clasificar. Es en el entorno militar donde se comienza a utilizar en el ámbito de la clasificación de víctimas en función de la urgencia requerida para su atención.

Una definición ampliamente aceptada es la siguiente: proceso simple, completo, objetivo y rápido de obtener una evaluación clínica inicial de víctimas con el objetivo de evaluar sus capacidades inmediatas de supervivencia y priorizarlas según su gravedad.

En situaciones críticas, el disponer de un método fiable y eficiente para la clasificación de víctimas es crucial. Generalmente los sistemas de triaje distinguen dos etapas:

- Primer triaje. Se lleva a cabo en la misma zona hostil. El personal a cargo del diagnóstico no debe pasar más de un minuto evaluando las capacidades de supervivencia de la víctima, ordenándolas finalmente de acuerdo a su gravedad. Algunos métodos en esta categoría son SHORT, START o MRCC.
- Segundo triaje. Esta etapa se desarrolla en una instalación sanitaria o un hospital. Aquí es el personal médico el que analizará el estado de la víctima: contusiones, heridas y lesiones. Algunos de los métodos aplicados en esta

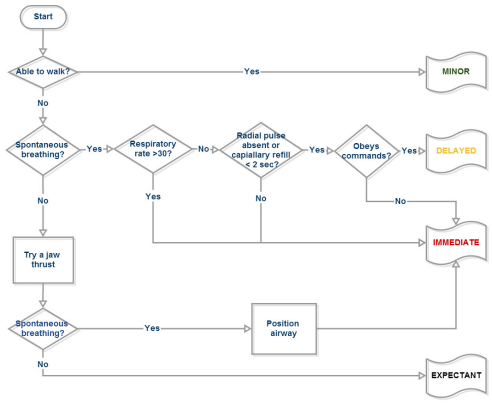


Figura 1. Algoritmo START

etapa son el Modelo Andorrano de Triaje (MAT), Spanish Triage System (SET) y Manchester Emergency Triage System (METS).

A pesar de realizar la distinción previa, estos métodos no son excluyentes sino complementarios. Los triajes correspondientes a la primera etapa proporcionan información indispensable para realizar la evacuación a las instalaciones sanitarias donde personal médico realizará la segunda etapa.

Se incluye en la figura 1 una descripción gráfica del método START. Éste es el método que se ha implementado en el sistema FastTriaje.

START utiliza etiquetas con cierto color para indicar el resultado de la clasificación, cada color representa un estado de gravedad diferente:

- negro: víctima mortal o irrecuperable.
- rojo: víctimas que requieren atención médica inmediata.
- amarillo: víctimas que requieren cuidados urgentes, pero su estado permite un retraso en la atención de entre media a una hora.
- verde: víctimas que no están seriamente heridas. Su tratamiento puede retrasarse más de una hora.

En el sistema propuesto en este trabajo las etiquetas tradicionales se sustituyen por etiquetas NFC ya que facilitan el tratamiento automático de la información generada en las valoraciones, permitiendo además que dicha información sea procesada fuera del lugar del suceso. En la sección siguiente se especifican algunas de las características que posee esta tecnología y que la hacen especialmente interesante para el sistema desarrollado.

III. Tecnología NFC

Near Field Communications (NFC) es es una tecnología de comunicación inalámbrica de alta frecuencia de corto alcance. En cierto modo se puede entender como una extensión de la tecnología RFID puesto que posibilita la coexistencia de los papeles de lector y tarjeta en un mismo dispositivo. Su principal funcionalidad es la de permitir la transferencia de contenido entre dispositivos móviles en modo punto a punto.

Una caraterística diferenciadora de esta tecnología frente a otras tales como RFID, Bluetooth, ZigBee o Wi-Fi, es que la transmisión de información en NFC no es continua, se requiere el contacto entre los dispositivos para el intercambio de información. Sin embargo dicha transferencia se realiza de manera rápida y oportuna.

Una de las principales ventajas de NFC cuando se compara con otras tecnologías es su seguridad inherente. Debido a su corto rango de comunicación y la necesidad de participación de los usuarios cuando se realiza una acción, NFC ofrece un nivel de seguridad más alto.

La tabla I recoge una comparación entre NFC y algunas tecnologías inalámbricas de comunicación.

Tabla II
Parámetros del campo de curvas elípticas en las ZKPs.

Parámetro	Descripción
p	número primo que define el campo F_p
a, b	coeficientes de la ecuación de la curva elíptica E
P	un punto base (un generador de un subgrupo cíclico de $E(F_p)$)
m	orden de P en $E(F_p)$

IV. Criptografía ligera y demostraciones de conocimiento nulo

Actualmente se diseñan aplicaciones móviles para casi cualquier ámbito de aplicación: negocios, gestión de transporte, redes sociales y muchos otros. En todos ellos el garantizar la seguridad de la información es una obligación. Las soluciones tradicionales generalmente requieren infraestructura específica, por lo que transferir estas soluciones al entorno aquí analizado no es viable [2].

En el sistema desarrollado se ha hecho uso de la criptografía ligera para garantizar el acceso legítimo a la información asociada a los triajes. Esta elección queda justificada por las restricciones sobre las capacidades computacionales y de comunicación definidas sobre los dispositivos que participan, generalmente teléfonos inteligentes. Concretamente, se usa criptografía de curvas elípticas (Elliptic Curve Cryptography, ECC) [3] debido a que:

- proporciona problemas con una complejidad computacional superior, y
- la longitud de clave que se necesita para alcanzar un nivel de seguridad concreto es más corta.

La tabla II describe la notación relacionada con las curvas necesaria para describir los protocolos implementados.

Las ZKPs permiten a un participante (el probador, A) convencer a otro (el verificador, B), sobre la veracidad de un hecho sin proporcionar más información que la validez de dicha demostración. Estas demostraciones se pueden extender para resolver el problema de autenticación tal y como se describe en el estándar ISO 9798-5 dedicado a autenticación de entidades.

Los elementos principales de las ZKPs son los siguientes tres:

Tabla I
COMPARACIÓN DE TECNOLOGÍAS INALÁMBRICAS DE CORTO ALCANCE.

	Tecnología			
Característica	NFC	Bluetooth	RFID	ZigBee
Establecimiento de conexión	0.1s o menos	6s	0.1s o menos	30ms
Velocidad	424-848kbps	24Mbps (versión 3.0)	424kbps	250kbps
Rango	10cm	10m	3m	70m
Consumo de batería	Bajo	Alto	Bajo	Bajo
Seguridad	Alta	Alta	Vulnerable	Vulnerable
Intervención de usuario	Tocar	Requiere configuración	Sin configuración	Sin configuración

Figura 2. Módulos de FastTriaje

- Testigo (w): el probador selecciona aleatoriamente un elemento de un conjunto predefinido manteniendo dicha elección en secreto. Este valor se denomina compromiso (x). A partir del mismo, se genera otro valor denominado testigo (w), siendo dicho valor remitido al verificador.
- Reto (e): En el segundo paso, el verificador selecciona aleatoriamente una pregunta que el probador debe responder correctamente, siempre y cuando realmente conozca la información secreta asociada al proceso de autenticación. Esta pregunta está relacionada con x y con las credenciales que deben ser verificadas.
- Respuesta (y): Finalmente el probador envía la respuesta al reto que será comprobada por el verificador. En caso de que la verificación sea correcta la autenticación se acepta.

V. EL SISTEMA DESARROLLADO: FASTTRIAJE

El principal escenario para el despliege de FastTriaje es una situación de emergencia o desastre natural. Su objetivo es agilizar la clasificación de víctimas y la gestión de la información generada durante ese proceso. La figura 2 ilustra los módulos que componen el sistema: la aplicación móvil y la plataforma web.

El objetivo principal de la aplicación Android es implementar el método de triaje START en dispositivos móviles como una herramienta simple, usable e intuitiva que facilita el proceso de triaje tradicional. Con dicha aplicación es posible enviar a una plataforma web los triajes realizados cuando el estado de las comunicaciones lo permitan. Actualmente la implementación desarrollada transfiere los triajes realizados haciendo uso de infraestructura Wi-Fi, pero es posible adaptarla para que use comunicaciones Wi-Fi Direct [4]. De esta manera, la transferencia de información se puede realizar directamente entre los dispositivos que forman parte de una red desplegada en la zona del desastre. Los usuarios registrados en el sistema podrán consultar posteriormente los triajes realizados. Además, cada triaje se puede almacenar en una etiqueta NFC que se anexa a la víctima para su clasificación "in situ". En cualquier caso la información asociada a cada triaje queda también registrada en el dispositivo móvil utilizado.

Sólo los dispositivos autorizados pueden intercambiar información con la plataforma web. Para garantizarlo se ha implementado una demostración de conocimiento nulo (Zero Knowledge Proof, ZKP) como protocolo de autenticación (ver ZKP1 en la siguiente sección). Aquellos usuarios que utilicen la aplicación deben estar registrados previamente en la plataforma web. Los privilegios de los usuarios, los establece el administrador de dicha plataforma. De este modo, se distinguirá entre usuarios con permiso sólo para consultar las etiquetas, de aquellos que pueden realizar los triajes y almacenarlos en las etiquetas NFC.

La plataforma web se comunica con la aplicación móvil a través de un servicio web. Antes de almacenar el triaje en la etiqueta NFC, el usuario autorizado ejecuta otra demostración de conocimiento nulo (ZKP2) que asocia el triaje con sus credenciales.

VI. MÉTODOS DE AUTENTICACIÓN EN FASTTRIAJE

Esta sección describe los protocolos de autenticación implementados en FastTriaje para garantizar el acceso sólo a los usuarios legítimos. Los dos protocolos comparten algunas características, principalmente en la etapa de inicialización. En ambos casos dicha etapa consiste en fijar una curva elíptica (E) y un punto base de la misma (P). Además las credenciales asociadas a A se definen de la misma manera: la identificación secreta es un entero seleccionado aleatoriamente del conjunto $\mathbb{Z}_p$, mientras que la pública es un punto de la curva E generado al multiplicar el entero asociado a la información secreta por el punto base.

VI-A. ZKP1: Autenticación del dispositivo móvil frente a la plataforma web

Este protocolo se utiliza para la autenticación del dispositivo móvil (A) frente a la plataforma web (B). Se incluye una descripción detallada del mismo en la tabla III. El reto definido en cada ejecución se genera a través de la aplicación de una función hash. Esta manera de proceder no se corresponde con la definición tradicional de las demostraciones de conocimiento nulo pero permite reducir el número de iteraciones del protocolo a sólo una.

Tabla III
ZKP1: AUTENTICACIÓN EL DISPOSITIVO MÓVIL FRENTE A LA PLATAFORMA WEB

Etapas	Accciones
Inicialización	número primo p E curva elíptica $\mathbb{Z}_p$ $P \in E$
Identificación secreta de A	$a \in \mathbb{Z}_p$
Identificación pública de A: $Puid_A$	$a * P \in E$
Compromiso:	$x \in_r \mathbb{Z}_p$
Testigo: $A \rightarrow B$	$w = x * P \in E$
Reto: $A \leftarrow B$	$e = hash(P, a * P, x * P)$
Respuesta: $A \rightarrow B$	$y = x + a * e \in \mathbb{Z}_p$
Verificación: B comprueba	$y * P - e * Puid_A = w$

Tabla IV
ZKP2: AUTENTICACIÓN DEL DISPOSITIVO MÓVIL FRENTE A LA ETIQUETA NFC.

Etapas	Acciones
Inicialización	p número primo E curva elíptica en $\mathbb{Z}_p$ $P \in E$
Identificación secreta de A	$a \in \mathbb{Z}_p$
Identificación pública de A: $Puid_A$	$a * P \in E$
Compromiso:	$\{x_1 * P, x_2 * P, \cdots, x_n * P\}$ $\in E$, con $x_i \in_r \mathbb{Z}_p$
Testigo: $A \rightarrow B$	$w = hash(x_j * P + x_k * P)$, con $j, k \in_r \{1, 2, \cdots, n\}$
Reto: $A \leftarrow B$	$e \in_r \mathbb{Z}_p$
Respuesta: $A \rightarrow B$	$y = x_j + x_k - a * e \in \mathbb{Z}_p$
Verificación: B comprueba	$hash(y * P - e * Puid_A)$ $= w$

VI-B. ZKP2: Autenticación del dispositivo móvil frente a la etiqueta NFC

El protocolo se destina a la autenticación del dispositivo móvil (A) frente a la etiqueta NFC antes de almacenar el triaje en ella.

Sin embargo, este protocolo no puede utilizarse directamente con las etiquetas NFC puesto que las etiquetas utilizadas en la implementación son totalmente pasivas. Esta elección se debe a intentar reducir costos de implementación lo que imposibilitaba la generación de retos por parte de la etiqueta. Por este motivo se optó por utilizar el paradigma de Fiat-Shamir [5] para transformar el protocolo propuesto generando una versión no interactiva.

De acuerdo con este paradigma se usa una función hash para la generación de los retos. En la implementación realizada se ha usado el nuevo estándar de función hash, SHA3 [6].

VII. CONCLUSIONES Y CUESTIONES FUTURAS

Este trabajo presenta un sistema que mejora la logística, clasificación y atención de víctimas en situaciones hostiles como pueden ser desastres naturales o accidentes. La herramienta está compuesta por una aplicación móvil y una plataforma web. La aplicación móvil sirve como asistente durante el desarrollo de los triajes y permite almacenar los resultados en etiquetas NFC que se asocian a las víctimas. Además es posible transferir el resultado de los triajes, a través de un servicio web, a una plataforma web donde la información generada puede procesarse de acuerdo con los perfiles de usuarios definidos.

Debido a que los servicios proporcionados se entienden como críticos, en la implementación realizada se ha tenido en cuenta la robustez del proceso de autenticación de entidades, así como la eficiencia del sistema.

Puesto que este trabajo está en desarrollo quedan algunas cuestiones importantes por solventar. Quizás una de las más significativas es dotar al sistema del servicio de confidencialidad. También caben mejoras a la hora de facilitar la coordinación e integración de los diferentes cuerpos de emergencia que pueden participar en la resolución de la misma. En este sentido, con la versión actual del sistema en el caso de que intervinieran diferentes cuerpos sanitarios habría que realizar la etapa de registro "in situ".

Otras cuestiones que se esperan incluir son las siguientes:

- Añadir funcionalidades estadísticas a la plataforma web.
- Integrar el sistema con las historias clínicas de los pacientes.
- Extender la aplicación para que posibilite la realización del segundo tipo de triajes.
- Se espera además ampliar la aplicación desarrollada completando el sistema de clasificación con extensiones específicas para grupos concretos de víctimas, tales como pacientes pediátricos y también con la implementación de sistemas de triajes pertenecientes a la categoría de segundo triaje.

AGRADECIMIENTOS

Este trabajo ha sido financiado parcialmente por el Ministerio de Economía y Competitividad y el Ministerio de Ciencia e Innovación bajo los proyectos IPT-2012-0585-370000: DEPHISIT y TIN2011-25452: TUERI.

REFERENCIAS

[1] K. V. Iserson and J. C. Moskop, "Triage in medicine, part i: Concept, history, and types," *Annals of Emergency Medicine*, vol. 49, no. 3, pp. 275 – 281, 2007.

[2] S. Babar, P. Mahalle, A. Stango, N. Prasad, and R. Prasad, "Proposed security model and threat taxonomy for the internet of things (IoT)," in *Recent Trends in Network Security and Applications*, ser. Communications in Computer and Information Science. Springer Berlin Heidelberg, 2010, vol. 89, pp. 420–429.

[3] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2003.

[4] J. I. S. González, "Implementación de algoritmos seguros en dispositivos wi-fi direct," 2013.

[5] A. Fiat and A. Shamir, “How to prove yourself: Practical solutions to identification and signature problems,” in *Advances in Cryptology - CRYPTO 86*. Springer-Verlag, 1987, pp. 186–194.
[6] G. Bertoni, J. Daemen, M. Peeters, and G. V. Assche, “ The Keccak sponge function family ,” Sooft.es, http://keccak.noekeon.org/papers.html.

La transformada de Walsh-Hadamard y otros parámetros en la autenticación biométrica

V. Gayoso Martínez, F. Hernández Álvarez, L. Hernández Encinas, F. Montoya Vitini y A. Orúe López
Departamento de Tratamiento de la Información y Criptografía
Instituto de Tecnologías Físicas y de la Información (ITEFI)
Consejo Superior de Investigaciones Científicas (CSIC)
C/ Serrano 144, 28006-Madrid, España
Emails: {victor.gayoso, fernando.hernandez, luis, fausto, amalia.orue}@iec.csic.es

***Resumen*—El patrón de iris es uno de los métodos biométricos más utilizados para la autenticación de individuos. No obstante, uno de sus principales desafíos consiste en lograr una baja tasa de falsos positivos (aceptación de un usuario ilegal) y de forma simultánea una baja tasa de falsos negativos (rechazo de un usuario legal), de modo que la seguridad del sistema sea la mayor posible. En este trabajo se presenta un primer estudio sobre la viabilidad de utilizar como método de verificación de una identidad basada en iris la transformada de Walsh-Hadamard, complementada con la covarianza cruzada y las distancias de Hamming y euclídea. Los primeros resultados muestran que la identificación basada en los cuatro parámetros anteriores presenta excelentes tasas de falsos positivos y negativos. Sin embargo, es preciso llevar a cabo estudios posteriores, que permitan ajustar mejor tales parámetros, para muestras con mayor número de usuarios.**

***Palabras clave*—Covarianza cruzada (Cross-covariance), Distancia de Hamming (Hamming distance), Distancia euclídea (Euclidean distance), Identificación biométrica (Biometric identification), Patrón de iris (Iris pattern), Transformada de Walsh-Hadamard (Walsh-Hadamard transform).**

I. Introducción

Como es sabido, la autenticación por métodos biométricos ([12], [13], [17]) consiste en verificar a individuos haciendo uso de alguno de sus rasgos fisiológicos como la cara ([6]), huella dactilar ([14]), impresión de la palma de la mano ([15]), iris ([18]), forma de la lengua ([10]), etc., o de comportamiento como la firma manuscrita ([5]), dinámica en la pulsación de teclas ([11]), olor y aroma de olor ([9]), etc.

Los sistemas biométricos ofrecen ventajas frente a otros métodos de autenticación, como los basados en contraseñas, dado que las características biométricas no se pueden perder u olvidar. Por otra parte, los rasgos biométricos son muy difíciles de copiar, falsificar, compartir o distribuir, y, además, requieren la presencia en el momento y en el lugar de quien se está identificando.

Sin embargo, a pesar de todas sus ventajas, el uso de los sistemas biométricos presenta algunos inconvenientes relacionados con la seguridad y la privacidad. Por ejemplo, algunos rasgos biométricos pueden ser grabados fácilmente sin el consentimiento del usuario, tales como la firma, voz, rostro, huella dactilar, etc. Además, a diferencia de lo que sucede con las contraseñas, PIN, etc., que se pueden renovar sin necesidad de que hayan sido comprometidos, los rasgos biométricos están asociados al usuario de forma permanente, de modo que si un rasgo se ve comprometido, no puede ser revocado o sustituido. Finalmente, si un rasgo biométrico se ve comprometido en una aplicación, todas las aplicaciones en la que este rasgo se utilice se verían comprometidas, por lo que dicho rasgo ya no será útil.

En general, el proceso para autenticar a un usuario por medio de su patrón biométrico consiste en dos fases: *inscripción* y *comprobación*. Durante la primera, se procesan por primera vez las plantillas biométricas y se almacenan en una base de datos (plantillas de referencia); mientras que en la segunda se extrae una nueva plantilla biométrica (llamada la plantilla de consulta) del usuario que quiere ser identificado y esta se compara con los datos ya almacenados (plantilla de referencia). Si la comparación es exitosa, el usuario queda autenticado; de lo contrario, su autenticación se rechaza.

El proceso de autenticación de usuarios puede llevarse a cabo de dos maneras, bien mediante una *verificación*, bien mediante una *identificación*. En el primer caso, se compara la plantilla del rasgo biométrico con la plantilla de referencia almacenada en la base de datos, es decir, el sistema realiza la comparación 1-a-1 para verificar la identidad del usuario. En la identificación, el objetivo es identificar una plantilla biométrica de un usuario desconocido como un individuo conocido dentro de un conjunto de los n posibles usuarios almacenados en una base de datos, esto es, la comparación es 1-a-n.

En general, los sistemas que utilizan un único patrón biométrico para la autenticación de individuos (unimodales) sólo disponen de la evidencia proporcionada por una única fuente de información, por lo que pueden plantear problemas relacionados con la variabilidad intra-usuarios e inter-usuarios (véase por ejemplo, [1]).

La variabilidad intra-usuarios hace referencia a las diferencias entre las plantillas de un mismo usuario extraídas en dos momentos distintos. Estas diferencias pueden causar el rechazo de un usuario legal si dos de sus plantillas son bastante diferentes (falso negativo). La variabilidad inter-usuarios se refiere a las similitudes que puede haber entre las plantillas de distintos usuarios. En este caso, tales similitudes pueden llevar a que el sistema acepte a un usuario ilegal (falso positivo).

Para paliar parte de los problemas mencionados más arriba se suelen utilizar sistemas multimodales, que utilizan varios

patrones biométricos simultáneamente.

En todo caso, existen dos coeficientes o tasas que permiten determinar la cantidad de falsos negativos o positivos que presenta un sistema de autenticación ([16]):

- *Tasa de falsa aceptación* (False Acceptance Rate, FAR). Este coeficiente determina la probabilidad de que el sistema considere una comparación positiva entre una plantilla de consulta y una plantilla de referencia en la base de datos que realmente no coinciden, esto es, es la probabilidad de que un usuario ilegal pueda, erróneamente, ser aceptado como un usuario conocido por el sistema (falso positivo). Esta tasa mide el porcentaje de coincidencias no válidas y es una medida relacionada con la seguridad del sistema.
- *Tasa de falso rechazo* (False Rejection Rate, FRR). Este valor calcula la probabilidad de que el sistema declare, incorrectamente, la no coincidencia entre la plantilla de consulta y la plantilla de referencia en la base de datos de un mismo usuario, es decir, es la probabilidad de que un usuario legal sea rechazado por el sistema (falso negativo). Esta tasa proporciona el porcentaje de entradas válidas que son rechazadas y es un criterio de comodidad.

Asociada a esta última, está la *tasa de aceptación genuina* (Genuine Acceptance Rate, GAR). Este valor es la probabilidad complementaria de la tasa de falso rechazo, es decir, es la probabilidad de que se considere correctamente a un usuario como usuario legal (verdaderos positivos). Esto es, $GAR = 1 - FRR$.

En este trabajo se presenta un primer estudio acerca de la viabilidad de utilizar como método de verificación de una identidad basada en iris la transformada de Walsh-Hadamard, complementada con la covarianza cruzada y las distancias de Hamming y euclídea. Para determinar su eficacia se calculan la tasa de falsa aceptación y la tasa de falso rechazo haciendo uso de un determinado número de las plantillas de iris empleadas en [7], donde se ha utilizado la base de datos de iris CASIA (Chinese Academy of Sciences' Institute of Automation) ([3]).

Se ha hecho uso de los cuatro parámetros mencionados más arriba debido a que los resultados de cada parámetro son diferentes, lo que permite ajustar las tasas mencionadas de forma más precisa. Se han descartado otras métricas (simetría, identidad, máximo de coincidencia de la varianza cruzada, etc.) porque no aportan mejoras con respecto a las consideradas finalmente, bien porque sus resultados ya estaban incluidos en alguno de los parámetros considerados, bien porque no discriminaban adecuadamente. Debe tenerse en cuenta que uno de los principales objetivos es lograr que la seguridad sea máxima, es decir, que la tasa de falsos positivos sea 0.

El resto de este trabajo se organiza de la siguiente manera. En la sección II se describe el algoritmo que se propone como método de verificación, señalando las propiedades de las cuatro medidas que se han empleado para la identificación de usuarios: la transformada de Walsh-Hadamard, la covarianza cruzada y las distancias de Hamming y euclídea. La sección III contiene los resultados experimentales que se han obtenido al ejecutar el algoritmo anterior con una muestra de plantillas de irises. Finalmente, las conclusiones y trabajos futuros de esta propuesta se incluyen en la sección IV.

II. Algoritmo de verificación de plantillas de irises

Las plantillas de irises que se han considerado proceden, en concreto, de la base de datos denominada *CASIA Iris Image Database Version 1.0* ([4]) que contiene 7 ficheros BMP (Windows bitmap) de 105 ojos, lo que contabiliza un total de 735 imágenes en escala de grises de 8 bits.

El procedimiento seguido en [7] para obtener las plantillas a partir de su imagen consta de los siguientes pasos:

1. Localización del iris y la pupila.
2. Identificación de los dos conos laterales del iris, descartando los conos superior e inferior a fin de evitar distorsiones producidas por las pestañas y los párpados.
3. Normalización de los conos laterales para obtener una imagen rectangular de 1024×128 bits.
4. División de la imagen en bloques de 32×32 bits, lo que genera un total de 32×4 bloques.
5. Análisis de cada bloque mediante filtros de Gabor con 4 orientaciones $(0, \pi/4, \pi/2, 3\pi/4)$ y 3 octavos en frecuencia. Cada orientación y frecuencia, aplicadas sobre cada bloque, genera dos bits.
6. Concatenación de los $32 \cdot 4 \cdot 4 \cdot 3 \cdot 2 = 3072$ bits que dan lugar al código asociado al iris.

La Figura 1 muestra un ejemplo del procesamiento de un iris, donde junto a los sectores laterales empleados en el cálculo puede observarse la imagen rectangular normalizada correspondiente.

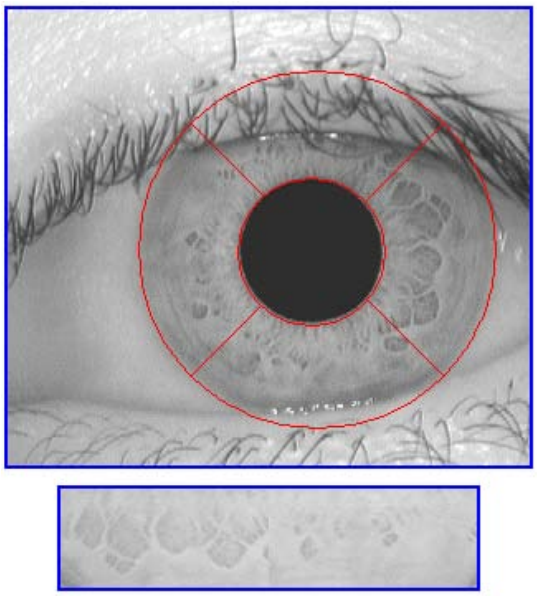


Figura 1. Ejemplo de generación de la plantilla asociada a un iris.

Los valores obtenidos se han almacenado en hexadecimal, conteniendo cada plantilla 384 bytes de información. La plantilla correspondiente a la imagen de la Figura 1 comienza de la siguiente manera:

```
9B47 CEB5 3D77 1E0C CB87 E41C 3736 9E0D
CF97 C51C 369B 8CC9 9666 665B 9A67 BA52
C466 374B 9B6D 9BD3 EC52 F74B DB6C AD92
                ...
```

En este trabajo se han programado cuatro sub-algoritmos de comparación de plantillas de iris: la distancia que proporciona la transformada de Walsh-Hadamard, la diferencia de la covarianza cruzada y las distancias de Hamming y euclídea.

Cada algoritmo suministra un valor de medida de proximidad, clasificándose el resultado como de *similitud* o *disimilitud* según que la medida arroje un resultado por encima, o por debajo, de un determinado valor de referencia elegido previamente.

La decisión de coincidencia de las plantillas de irises se toma en base a los resultados de los cuatro algoritmos de la siguiente forma: un usuario es aceptado si en alguno de los cuatro algoritmos es considerado como similar; mientras que es rechazado si es disimilar para todos ellos.

II-A. Diferencia de la Transformada de Walsh-Hadamard

La Transformada de Walsh-Hadamard (WHT) es una transformada ortogonal, similar a la transformada de Fourier, que hace corresponder a una secuencia numérica otra secuencia formada por funciones de Walsh, en lugar de funciones sinusoidales ([8]). Las funciones de Walsh solo tienen valores $+1$ y -1 y por tanto resulta la más adecuadas para transformaciones de secuencias discretas de números, mientras que la transformada de Fourier es óptima para señales continuas. La WHT es más rápida si se calcula con 512 puntos y sus resultados no mejoran calculando más puntos.

La WHT ha sido propuesta para ser empleada en la selección de características faciales ([2]). Aquí se propone su uso como un medio para la obtención de un parámetro que permita decidir si dos plantillas de irises son o no similares. A modo de ejemplo, en la Figura 2 se ilustra la WHT de la plantilla del iris del usuario 1 de la base de datos CASIA.

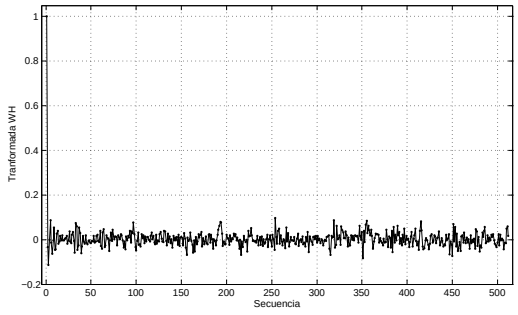


Figura 2. WHT de la plantilla del iris del usuario 1.

La Figura 3 representa la diferencia entre las transformadas de dos plantillas diferentes del iris del mismo usuario. El sub-algoritmo utilizado en este caso, consiste en calcular la diferencia cuadrática media de las secuencias de la WHT de dos irises diferentes, sean o no del mismo usuario.

La Figura 4 representa la diferencia entre las WHT de dos plantillas de irises de diferentes usuarios.

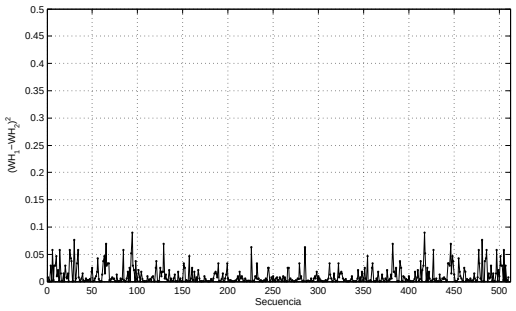


Figura 3. Diferencia entre WHT de dos plantillas del iris del usuario 1.

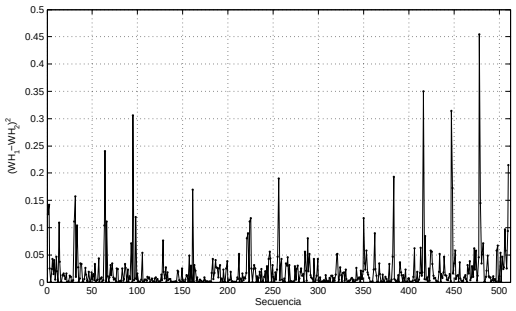


Figura 4. Diferencia entre WHT de las plantillas de los usuarios 1 y 2.

Se puede apreciar que la amplitud de la diferencia de términos de dos irises del mismo usuario es mucho menor, en conjunto, que la diferencia de términos de dos irises de distintos usuarios. Experimentalmente se ha encontrado que el valor de referencia óptimo para la diferencia cuadrática media de las secuencias de la transformada es $WH = 0{,}004$, clasificándose como *similares* los irises con valores medios menores y como *disimilares* los irises con valores mayores o iguales.

II-B. Diferencia de la covarianza cruzada

La covarianza es un valor que indica el grado de variación conjunta de dos variables aleatorias. Es el dato básico para determinar si existe una dependencia entre ambas variables. Cuando las dos variables son idénticas se denomina auto-covarianza y si son diferentes es la llamada covarianza cruzada. La Figura 5 ilustra la auto-covarianza de un patrón del iris del usuario 1 de la base de datos CASIA, normalizada para que el valor máximo sea 1.

El sub-algoritmo de comparación de irises utilizado es el siguiente:

- En primer lugar se calcula la auto-covarianza de la plantilla de un iris de determinado usuario.
- A continuación se determina la covarianza cruzada entre la misma plantilla y otra plantilla diferente (la que se desea comparar con la anterior).
- Más tarde se halla la diferencia entre ellos, término a término.
- Finalmente, se calcula la media cuadrática de estas diferencias.

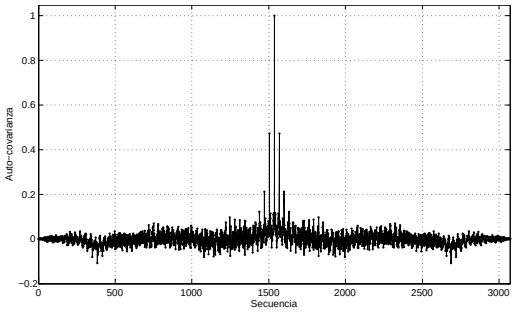


Figura 5. Auto-covarianza de la plantilla del usuario 1.

La Figura 6 ilustra la covarianza cruzada entre dos plantillas diferentes del iris del usuario 1 de la base de datos CASIA. La Figura 7 ilustra la covarianza cruzada de una de las plantillas de iris del usuario 1 y otra del usuario 2. Se puede apreciar que la covarianza cruzada de dos irises del mismo usuario es mucho menor, en conjunto, que la covarianza cruzada de patrones de dos irises de distintos usuarios.

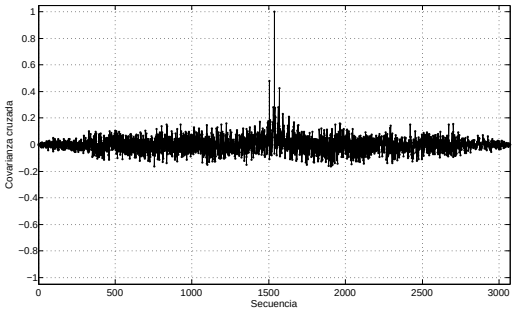


Figura 6. Covarianza cruzada de dos plantillas del usuario 1.

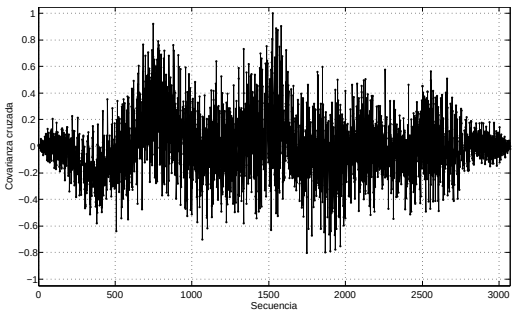


Figura 7. Covarianza cruzada de las plantillas de los usuarios 1 y 2.

Experimentalmente se ha encontrado que, dados los valores de prueba considerados, el valor de referencia óptimo para la diferencia cuadrática media de las covarianzas cruzadas de las secuencias de la transformada es $Xv = 0{,}01$, clasificándose como *similares* los irises con valor de esta diferencia menor que el valor de referencia y como *disimilares* los irises con valores mayores o iguales.

II-C. Distancia de Hamming

Para comparar dos archivos de plantillas de irises de la misma longitud de m muestras, se determina el valor medio de las distancias de Hamming entre las muestras que ocupan el mismo lugar en cada archivo.

La distancia de Hamming dh se ha determinado contando la cantidad de bits idénticos en ambas muestras. Se han utilizado muestras codificadas con 2 bits, por lo que esta distancia puede ser 0, 1 o 2. Experimentalmente se ha encontrado que un valor de referencia con buenos resultados es $DH = 0{,}5$. Así, se clasifican como *similares* las plantillas de irises cuyo valor medio de distancias de Hamming son menores que dicho valor, $dh < DH$, y como *disimilares* las plantillas con valores iguales o mayores que el dado, $dh \geq DH$.

II-D. Distancia de euclídea

Consiste en un sub-algoritmo similar al anterior, que en lugar de calcular la distancia de Hamming entre las muestras de las plantillas de irises calcula la media de la diferencia euclídea de los valores absolutos de las muestras de los patrones, que puede variar, en decimal, entre 0 y 3.

Experimentalmente se ha encontrado que el valor de referencia óptimo es $DE = 1$, clasificándose como *similares* los iris con valor medio de distancias euclídeas menores que este valor y como *disimilares* los iris con valores iguales o mayores que el de referencia.

III. Resultados experimentales

En la parte experimental se han considerado las plantillas de los 7 irises de 105 individuos de la base de datos CASIA ([3]). Estas 735 plantillas han servido como base de datos para contrastar el rendimiento del algoritmo presentado en la sección II.

Dado que se trata de analizar los valores de las tasas de falsa aceptación (*FAR*) y falso rechazo (*FRR*), se ha ejecutado el algoritmo presentado en la sección II de modo que cada una de las 7 plantillas de irises de cada uno de los 105 usuarios se ha comparado con las $735\ (= 105 \cdot 7)$ plantillas de la base de datos, obteniéndose una tabla de tamaño 735×735 (se omite la presentación de esta tabla por razones de espacio).

Para el estudio de la variabilidad intra-usuarios, cada una de las 7 plantillas de los 105 usuarios se considera como la entrada de la fase de verificación y se compara con el resto de las plantillas del mismo usuario. El resultado de esta comparación muestra el nivel de similitud entre todas las plantillas de un único usuario. El número de similitudes permite medir la tasa de falso rechazo. Así pues, si se consideran todas las comparaciones de un usuario consigo mismo se obtienen $49\ (= 7 \cdot 7)$ comparaciones, de modo que el número total de comparaciones es de $5145\ (= 49 \cdot 105)$. En el experimento realizado se ha obtenido que las comparaciones exitosas entre los 105 usuarios es el siguiente valor:

$$\mathit{GAR} = \frac{4091}{5145} \approx 0{,}7951 \equiv 79{,}51\,\%.$$

Por tanto, se tiene que la tasa de falso rechazo, es decir, los falsos negativos son:

$$FRR = 1 - GAR \approx 1 - 0{,}7951 = 0{,}2049 \equiv 20{,}49\,\%.$$

Considerando cada uno de los parámetros por separado, los resultados que se han obtenido son los siguientes: la distancia euclídea proporciona un $65{,}34\,\%$ de verdaderos positivos, la distancia de Hamming un $71{,}21\,\%$, la covarianza cruzada un $71{,}43\,\%$ y la WHT un $69{,}05\,\%$; mientras que considerando todas juntas, el resultado es del $79{,}51\,\%$, lo que supone una ganancia considerable. Además, ninguno de los parámetros proporciona falsos positivos.

Como era de esperar, la aportación a la verificación de cada uno de los parámetros es diferente. Así, si no se considera alguno de los parámetros, la tasa de verdaderos positivos disminuye, especialmente si no se considera la distancia de WHT, en cuyo caso los verdaderos positivos serían solamente del $76{,}95\,\%$. Por tanto, es necesario incluir esta transformada entre los parámetros de discriminación para obtener mejores resultados, aunque su coste computacional sea el más elevado.

En el estudio de la variabilidad inter-usuarios, se compara cada una de las 7 plantillas de cada uno de los 105 usuarios con las 7 plantillas de los restantes 104 usuarios y se determina su similitud o disimilitud. Dado que hay un total de 535080 $(= 7 \cdot 105 \cdot 7 \cdot 104)$ comparaciones y no hay disimilitudes, la tasa de falsa aceptación, es decir, los falsos positivos son:

$$FAR = \frac{0}{535080} = 0{,}0 \equiv 0\,\%.$$

Finalmente, el coste computacional, para comparación, del algoritmo de la distancia euclídea es de $0{,}19$ ms, de la distancia de Hamming es $0{,}55$ ms, de la varianza cruzada es $0{,}87$ ms y de la WHT es $5{,}22$ ms. La comparación de una plantilla contra las 735 de la base de datos requiere $4{,}7$ segundos. Debe tenerse en cuenta que los algoritmos se han ejecutado bajo MatLab en un PC de 2 GHz, por lo que sería posible obtener mejores resultados si estos se implementaran en C, por ejemplo.

IV. Conclusiones

Con el fin de mejorar las tasas de falsos positivos y falsos negativos en la identificación de usuarios mediante plantillas de irises, se ha propuesto el uso de un algoritmo que considere cuatro parámetros derivados de las distancias de la transformada de Walsh-Hadamard y de la covarianza cruzada, así como de las distancias de Hamming y euclídea.

Este algoritmo considera que dos plantillas de irises son similares, y por tanto que ambas pertenecen a un mismo individuo, si alguno de los cuatro parámetros anteriores declaran ambas plantillas como similares. En caso contrario, esto es, si ninguno de los cuatro parámetros lo considera similar, las plantillas se consideran disimilares y la identificación es rechazada.

El algoritmo, en su versión actual, permite utilizar una única fuente de información (unimodal), proporcionando una tasa de falsos negativos del $20{,}49\,\%$ y de faltos positivos del $0\,\%$. Esto es, según el algoritmo propuesto y con la muestra de usuarios empleada, no se aceptan individuos ilegales ($FAR = 0{,}0$) a la vez que el porcentaje de individuos legales que son erróneamente rechazados es cercano al $20\,\%$ ($FRR = 0{,}2049$).

A la vista de los resultados obtenidos, es necesario incluir la transformada de Walsh-Hadamard entre los parámetros del algoritmo para mejorar los resultados, aunque su coste computación sea el más elevado.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el Ministerio de Ciencia e Innovación (España) bajo el proyecto TIN2011-22668.

Referencias

[1] R. Alvarez Marino, F. Hernandez Alvarez, and L. Hernandez Encinas, "A crypto-biometric scheme based on iris-templates with fuzzy extractors," *Information Sciences*, vol. 195, pp. 91–102, 2012, http://dx.doi.org/10.1016/j.ins.2012.01.042.

[2] W. Besbas, M. Artemi, R. Sullivan, and M. Al Rjebi, "Content based face image retrieval in Walsh Hadamard transform domain," in *The International Conference on Computing, Networking and Digital Technologies (ICCNDT2012)*, 2012, pp. 101–106.

[3] Biometric Ideal Test, "CASIA iris image database," 2010, http://biometrics.idealtest.org/findDownloadDbByMode.do?mode=Iris.

[4] ——, "CASIA iris image database, version 1.0," 2010, http://www.idealtest.org/dbDetailForUser.do?id=1.

[5] R. Blanco-Gonzalo, O. Miguel-Hurtado, A. Mendaza-Ormaza, and R. Sanchez-Reillo, "Handwritten signature recognition in mobile scenarios: Performance evaluation," in *2012 IEEE International Carnahan Conference on Security Technology (ICCST'2012)*, 2012, pp. 174–179.

[6] J. Connolly, E. Granger, and R. Sabourin, "An adaptive classification system for video-based face recognition," *Information Sciences*, vol. 192, no. 1, pp. 50–70., 2012.

[7] E. Diez Laiz and C. Sanchez Avila, "Sistema criptobiométrico basado en iris para esquemas Diffie-Hellman con curva elíptica (ECDH)," in *Congreso de Métodos Numéricos en Ingeniería*, 2009, pp. 1–20.

[8] D. Elliot and K. Rao, *Fast transforms, algorithms, analysis, applications*. New York: Academic Press, 1982.

[9] V. Fernandez Mateos, F. Hernandez Alvarez, L. Hernandez Encinas, C. Sanchez Avila, and G. Bailador, "Towards a biometric identification based on corporal odor," in 4^{th} *International Information Security & Cryptology Conference (ISCTURKEY'10)*, May 2010.

[10] B. Huang, J. Wu, Z. Zhang, and N. Li, "Tongue shape classification by geometric features," *Information Sciences*, vol. 180, pp. 312–324, 2010.

[11] J. Ilonen, "Keystroke dynamics," 2013, http://www.it.lut.fi/kurssit/03-04/010970000/seminars/Ilonen.pdf.

[12] A. Jain, R. Bolle, and S. Pankanti, *Biometrics: Personal Identification in Networked Society*. New York: Springer, 1999.

[13] A. Jain, A. Ross, and S. Pankanti, "Biometrics: A tool for information security," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 125–143, 2006.

[14] A. Jain, A. Ross, and S. Prabhakar, "Fingerprint matching using minutiae and texture features," in *International Conference on Image Processing (ICIP)*, 2001, pp. 282–285.

[15] H. Li, J. Zhang, and Z. Zhang, "Generating cancelable palmprint templates via coupled nonlinear dynamic filters and multiple orientation palmcodes," *Information Sciences*, vol. 180, pp. 3876–3893, 2010.

[16] J. Mainguet, "Biometrics," 2013, http://pagesperso-orange.fr/fingerchip/biometrics/biometrics.htm.

[17] D. Maltoni, D. Maio, A. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. New York: Springer, 2003.

[18] C. Sanchez Avila and R. Sanchez-Reillo, "Two different approaches for iris recognition using Gabor filters and multiscale zero-crossing representation," *Pattern Recognition*, vol. 38, no. 2, pp. 231–240, 2005.

Hacia un Proceso de Migración de la Seguridad de Sistemas heredados al Cloud

Luis Márquez Alcañiz
Comision Nacional
de la Competencia
Madrid
luismarquezalcaniz@gmail.com

David G. Rosado
Departamento de TSI
Grupo GSyA
Universidad Castilla-La Mancha
Ciudad Real
david.grosado@uclm.es

Daniel Mellado
Agencia Tributaria
Madrid
damefe@esdebian.org

Eduardo Fernández-Medina
Departamento de TSI
Grupo GSyA
Universidad Castilla-La Mancha
Ciudad Real
eduardo.fdezmedina@uclm.es

***Resumen*—El desarrollo de la computación en la nube es una tendencia fuerte en la industria de las TI que hace que los clientes de este nuevo modelo de prestación de servicios, sobre todo las empresas, se enfrenten a desafíos nuevos en lo que se refiere a la gestión de la seguridad de sus aplicaciones heredadas en el nuevo entorno. La cuestión es en cómo migrar de forma segura los sistemas de información heredados de estas empresas. Este artículo presenta un proceso (SMiLe2Cloud) y un marco de trabajo con el que se puede migrar de forma segura los sistemas corporativos heredados a infraestructuras o entornos en la nube, siguiendo los 14 dominios de seguridad del CSA y utilizando ingeniería inversa.**

***Palabras clave*—Cloud, seguridad informática, migración de sistemas heredados, KDM, SLA, SecSLA.**

I. Introducción

Para algunos expertos, la computación en la nube está "desalineada con los modelos y controles de seguridad tradicionales" [1]. Sin embargo, otros ven en este modelo una gran oportunidad para mejorar la seguridad de los sistemas heredados [2]. Sin embargo, hay algo en lo que todos coinciden: la nube supone nuevas amenazas y estas amenazas deben ser resueltas antes de que las aplicaciones de las grandes corporaciones entren en juego.

¿Qué tienen en especial esas aplicaciones de las grandes corporaciones? Que la mayoría de ellas se basan en sistemas de información heredados (LIS-Legacy Information Systems). Según una encuesta realizada por MeriTalk [3] a un total de 166 directivos de TI del gobierno federal norteamericano, el 47 % de las aplicaciones de TI se basan en tecnología heredada que necesita modernización?. Y gran parte de la modernización no sólo se beneficiaría de una mejora tecnológica pura, sino que entrarían en juego reducciones de coste importantes a raíz de una migración a la nube de parte de la infraestructura que las soporta [4].

Y sin embargo, aunque la modernización de los LIS por medio de la migración a la nube podría implicar inmensos ahorros y reducciones de los presupuestos, y a pesar de la preocupación a la que antes nos hemos referido relativa a la seguridad intrínseca del modelo en la nube, hasta la fecha parece que todavía no hay un modelo que permita la migración a la nube de sistemas que de forma explícita incluyan procesos relacionados con la seguridad de dichos sistemas. Sí que es cierto que existen propuestas de procesos de migración [5][6][7][8], pero ninguno de ellos propone una verdadera integración con las cuestiones específicas de seguridad en forma de necesidades y/o de oportunidades que se derivan del modelo en la nube.

Nuestro propósito con este artículo es proponer un marco de trabajo para tal proceso mediante un conjunto de métodos que resuelvan de forma concreta las cuestiones de seguridad y la integración de la seguridad con procesos de otra naturaleza orientados todos ellos a la migración segura a la nube de sistemas de información heredados. En [9] se realizó un estudio de la importancia de la seguridad en los entornos Cloud y se analizó algunas propuestas de migración al Cloud, que fue descrito formalmente planteando un "mapping study" en [11], donde se indica la falta de iniciativas con respecto a la seguridad en el propio proceso de migración. En [10] se da algunas pautas y criterios a la hora de tomar algunas decisiones en cuanto a qué características debemos migrar al Cloud y cuáles no. Este artículo, que se presenta aquí, avanza en el sentido de que una vez descubierta la necesidad de disponer de un proceso de migración donde se incorpore la seguridad desde el principio, se define dicho proceso de migración con el propósito de servir de soporte y ayuda para migrar las características de seguridad de sistemas heredados al Cloud Computing.

El artículo está estructurado en 2 secciones adicionales a esta introducción. En la sección 2 presentamos el marco de trabajo propiamente dicho. Y en la sección 3 ofrecemos unas someras conclusiones y presentamos lo que serán las líneas de actuación futuras.

II. SMILE2CLOUD: Proceso para la Migración a la nube de la Seguridad de los Sistemas Heredados

En esta sección proponemos un proceso (denominado SMiLe2Cloud - Security MIgration of LEgacy systems TO Cloud computing) que pretende resolver el problema de la migración con seguridad a la nube de sistemas de información heredados. Este proceso está basado en el modelo de herradura del SEI (Software Engineering Institute) [12], pero también tiene una vocación de proceso de mejora continua al estilo de Deming.

Dado que estamos interesados en los aspectos propiamente relacionados con la seguridad (y no en los esfuerzos generales de ingeniería inversa necesarios para obtener la especificación funcional) hemos partido de la base de que los ingenieros a cargo de la migración ya han desarrollado un modelo del sistema heredado que define las especificaciones funcionales y los elementos arquitectónicos de sistema (con exclusión de las especificaciones relacionadas con la seguridad y la arquitectura de seguridad) y que han documentado dichas especificaciones y elementos en un entorno que puede exportar dicha especificación en formato KDM (Knowledge Discovery Metamodel) [13]. Es en este punto en el que nosotros entramos y empezamos a desarrollar los aspectos de seguridad a partir del diseño obtenido mediante ingeniería inversa y luego continuamos con el resto del proceso de segurización del sistema migrado.

II-A. Visión general

Como se ha indicado antes, nuestro proceso comienza en el punto más alto del modelo de herradura del SEI, una vez que la arquitectura básica ha sido obtenida, y justo antes de que comience la transformación. Desde este punto, continuará transformando y refinando el sistema objetivo, ya desde una perspectiva puramente enfocada en los temas relacionados específicamente con la nube.

El proceso SMiLe2Cloud consta de siete actividades dirigidas por 14 dominios de seguridad del CSA (Cloud Security Alliance) [14] que son mostradas en Figura 1. La actividad de *"extracción"* está enfocada al uso de la reingeniería inversa para extraer aspectos de seguridad desde el LIS a un modelo de seguridad (modelo SMiLe) definido para nuestro proceso de migración. La segunda actividad es la *"valoración"* durante la cual se estudian las principales características del cloud, los principales proveedores y diferentes modelos cloud. La tercera actividad es el *"análisis"* de los requisitos de seguridad, las cláusulas en los acuerdos a nivel de servicio de seguridad y los servicios ofrecidos por los proveedores de seguridad del cloud. La actividad de *"diseño"* está enfocada en el diseño de la arquitectura de seguridad y en la definición de una estrategia de migración que será aplicada en la siguiente actividad del proceso de migración, que es la actividad de *"migración"* donde los elementos de seguridad son desarrollados, configurados y contratados siguiente la estrategia previamente definida. La sexta actividad es la *"evaluación"* donde se verifica y valida el modelo de seguridad migrado. Finalmente, la actividad de *"mejora"* captura los nuevos aspectos de seguridad que se quieren incorporar dentro de un nuevo ciclo del proceso y se analizan las mejoras y cambios propuestos para nuestro sistema cloud.

Dado que KDM carece de elementos específicos para modelar aspectos de seguridad de un sistema heredado, en realidad parte de nuestro proceso debe realizarse antes de que exista una especificación completa del sistema obtenida por ingeniería inversa. La actividad de extracción, específicamente definida en nuestro proceso, precisamente trata con esta última parte de la fase de reingeniería del modelo de herradura. Sin embargo, esta fase no es específica de un proceso de migración a la nube. Podría ser utilizada de forma separada en cualquier proceso que pretendiera migrar un sistema heredado de forma segura a cualquier tipo de arquitectura objetivo.

Lo que sí es necesario entender de antemano, cuando estamos pensando en migrar a la nube, es el papel central que tienen para la seguridad y para la arquitectura del sistema completo los acuerdos de nivel de servicio (SLA - Service Level Agreement) específicos de seguridad (comúnmente denominados SecSLA). Los SecSLA son el núcleo de la seguridad en la nube y la mayoría de controles específicos que se pueden implantar se instancian como cláusulas en el SecSLA siempre que es posible. Por supuesto, esto depende en gran media del modelo de despliegue elegido; con un modelo de infraestructura como servicio (IaaS - Infrastructure as a Service) como el que ofrece Amazon EC2, la organización que está migrando el sistema heredado tiene que trabajar a un nivel más bajo y diseñar e implementar controles tradicionales por sí misma; sin embargo, con modelos de software como servicio puros (SaaS - Software as a Service), casi todos los controles de seguridad deben ser implementados como SecSLA ya sean acordados con el proveedor funcional del servicio o con un proveedor específico de seguridad como servicio (SecaaS - Security as a Service); finalmente con un modelo de plataforma como servicio (PaaS - Platform as a Service) como el que ofrece Google App Engine una solución intermedia que balancee controles de ambos tipos será la aproximación adecuada (la seguridad de la plataforma recae en el proveedor y la seguridad de las aplicaciones y la seguridad del propio proceso de desarrollo y despliegue es responsabilidad del cliente).

Todo esto es importante para la definición de la arquitectura de seguridad, puesto que algunas actividades en un proceso tradicional de aseguramiento de sistemas (ya sea en migración de sistemas o en desarrollo de sistemas desde cero) implican el diseño de controles, mientras que en un proceso orientado a la nube, la mayoría del proceso tiene que ver con el aspecto nuclear de seleccionar qué controles diseñados por los proveedores son aplicables y asegurar que las cláusulas del SLA cubren dichos controles. De esa manera, las cláusulas se convierten, de facto, en los propios controles que salvaguardan a la organización cliente (normalmente mediante la aplicación de obligaciones contractuales o penalizaciones en caso de que el proveedor no pueda cumplir dichas obligaciones). El problema, pues, se convierte en una mezcla de diseño de sistemas, selección de proveedores de servicio y técnicas de negociación de contratos.

En nuestro caso, el objetivo es orientar nuestra aproximación lo más posible hacia la ingeniería de sistemas de información. Por ello excluiremos inicialmente las soluciones puramente SaaS que tienden a estar orientadas principalmente hacia la reingeniería de procesos que a la de sistemas. Esto es, una propuesta SaaS supone normalmente un diseño de cómo el proceso de negocio debe ser migrado (es decir cómo podemos seleccionar el mejor proveedor SaaS que pueda cumplir con el proceso de negocio y/o en qué manera debe cambiar dicho

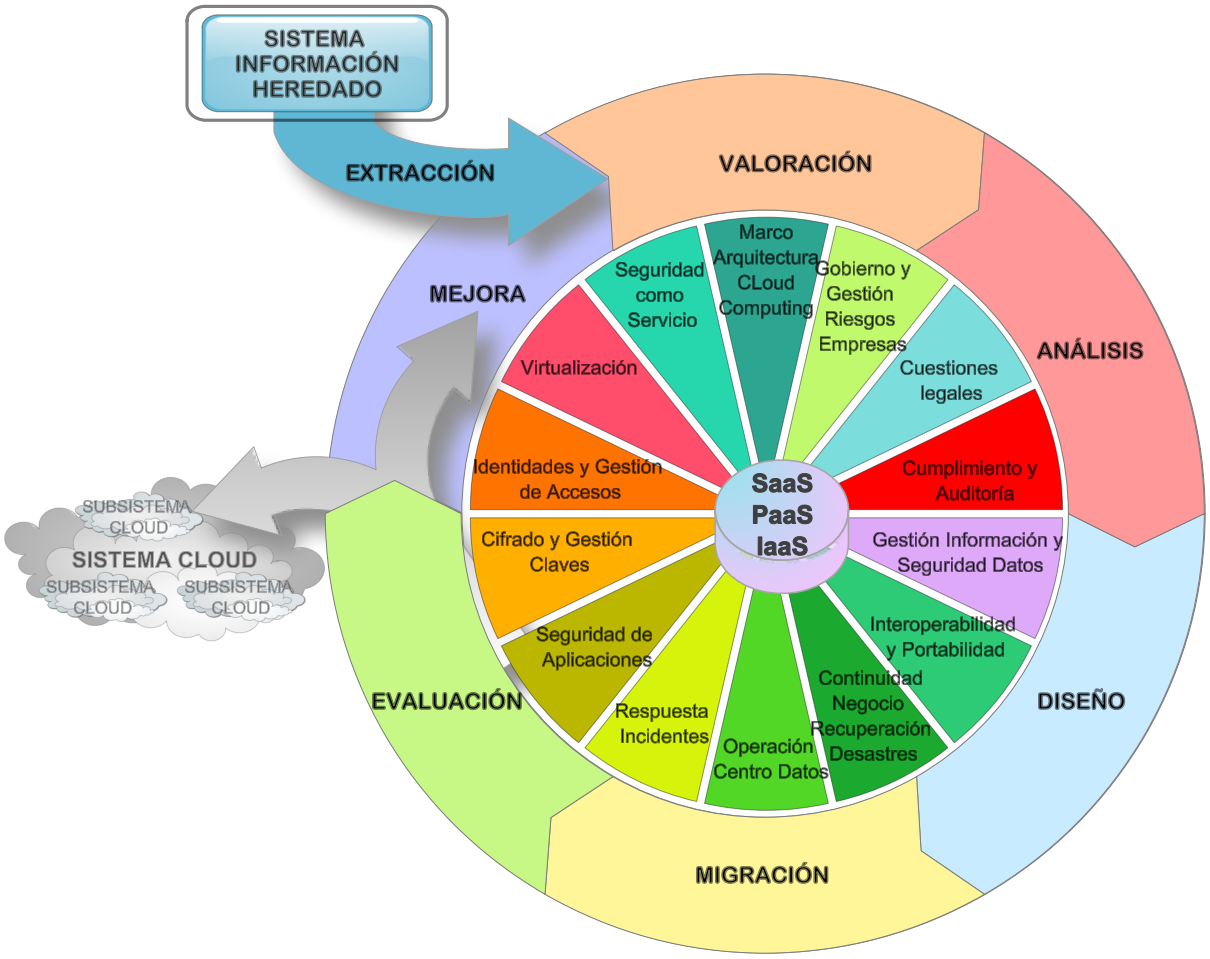


Figura 1. El proceso SMiLe2Cloud: un proceso para migrar a la nube la seguridad de sistemas de información heredados.

proceso de negocio para acomodarse al nuevo sistema) pero tiene poco que ver con cuestiones relacionadas con la ingeniería de sistemas. En cierto sentido, una solución puramente SaaS no sería una migración pura de un sistema heredado, sino que sería un cambio completo del sistema que trataría con cuestiones como la migración de los datos del sistema heredado original más que la migración de funcionalidades.

II-B. Actividades de SMiLe2Cloud

En esta sección presentaremos una descripción en profundidad del conjunto de actividades en nuestro proceso SMiLe2Cloud las cuales son mostradas en Figura 1. El proceso tiene 7 actividades: Extracción, Valoración, Análisis, Diseño, Migración, Evaluación y Mejora, y un amplio conjunto de artefactos de entrada y salida para cada una de las actividades y que son descritas de forma resumida a continuación.

II-B1. Actividad 1: Extracción: La extracción es la actividad en la que el modelo de seguridad del sistema heredado es obtenido a partir del propio código del sistema y de la documentación del mismo. Se trata de un subproceso de ingeniería inversa que se puede realizar en paralelo al subproceso de obtención del modelo de arquitectura general del sistema heredado. Normalmente ambos procesos se supone que son realizados con la ayuda parcial de herramientas de ingeniería inversa que faciliten las tareas y pasos que el analista debe realizar para identificar los diferentes requisitos y controles de seguridad existentes en el sistema origen.

Se trata de una actividad orientada por los datos y parte de la especificación formal de los programas y subprogramas del sistema heredado, así como de los datos gestionados por cada unidad de programa. Esta especificación formal tiene la forma de árbol de sintaxis abstracta (AST - Abstract Syntax Tree) que modeliza cada unidad de programa y los datos manejados.

- A1.1 Definir el árbol de sintaxis abstracta (AST-abstract syntax tree)

Un árbol de sintaxis abstracta es una representación en forma de árbol de la estructura del programa y de los elementos de datos del sistema heredado y ofrece una equivalencia 1-a-1 entre todos los elementos incluidos en el código en forma de estructura arbórea. El AST es usado para derivar los requisitos de seguridad del sistema.

- A1.2 Extraer aspectos de seguridad del AST

Para cada elemento de datos y de subprograma que ha sido representado en el AST, el analista de sistemas debe extraer los parámetros de seguridad concretos para cada uno de los perfiles de usuarios definidos en su operación habitual normal (acceso, creación, modificación, borrado, administración, auditoría).

- A1.3 Definir el modelo de seguridad en KDM (Knowledge Discovery Metamodel)

Nuestra aproximación propone evitar esta situación haciendo que cada artefacto y control de seguridad del sistema heredado sea instanciado en una regla de seguridad de negocio y se incluye en el modelo conceptual durante la fase de análisis.

- A1.4 Definir el modelo de seguridad (modelo SMiLe)

El modelo SMiLe (Security MIgration of LEgacy systems) es un modelo de seguridad de un sistema heredado que ha sido derivado desde las reglas de negocio de seguridad definidos mediante KDM y los activos identificados en el paso A1.3. Ahora es necesario incluir las políticas y controles de seguridad que fueron predefinidos para el sistema heredado (con independencia de si el sistema debe ser migrado a la nube o no).

II-B2. Actividad 2: Valoración: La actividad de valoración es en la que el modelo general de seguridad del sistema heredado es adaptado al nuevo entorno (en nuestro caso, a la nube). Comenzamos con un modelo SMiLe que no está específicamente adaptado al entorno de la nube y en dicho modelo estudiamos las fortalezas, debilidades, oportunidades y amenazas específicas que la nube incorpora. Esta actividad comienza con el modelo SMiLe (esto es, el modelo de seguridad del sistema heredado obtenido por ingeniería inversa) y es realmente la primera actividad de ingeniería directa del modelo de herradura que define nuestro proceso.

Los objetivos de esta actividad son los siguientes: refinar el modelo SMiLe para obtener un modelo SMiLe2Cloud (esto es, adaptar el modelo del sistema heredado con las amenazas específicas de la nube, los activos específicos en la nube, los escenarios específicos de la nube, los requisitos específicos de la nube, etc.); seleccionar un conjunto de proveedores de servicios en la nube y de proveedores de seguridad en la nube que, al menos parcialmente, cumplan con los requisitos de seguridad del modelo SMiLe2Cloud del sistema heredado según nuestra especificación de seguridad; y validar los modelos de servicio y de despliegue que pueden utilizarse dentro de los límites de dichas especificaciones de requisitos de seguridad.

- A2.1 Definir la matriz DAFO (Debilidades, Amenazas, Fortalezas y Oportunidades) e incorporar los elementos específicos de la nube en el modelo SMiLe

Se define una matriz (DAFO) con las debilidades, fortalezas, oportunidades y nuevas amenazas que el modelo cloud plantea al LIS.

- A2.2 Validar proveedores en la nube

Una vez que la matriz DAFO se ha completado, el analista debe contrastarlo con el modelo SMiLe del LIS y comprobar la lista de proveedores de servicios cloud que puede abordar las especificaciones funcionales del LIS y extraer las especificaciones de seguridad que ofrecen dentro de los términos del acuerdo de nivel de servicio. El analista también debe comprobar cuáles términos relacionados con la seguridad del acuerdo a nivel de servicio están abiertos a negociación.

- A2.3 Validar modelos en la nube

Dado que las diferentes propuestas de modelos cloud (modelos de servicios y modelos de despliegue) forman parte de la arquitectura del modelo cloud y no del modelo de seguridad, no se debe tratar de cambiar los modelos seleccionados o propuestos definidos en la arquitectura LIS. Sin embargo, los modelos conducen a una diferencia en las restricciones de seguridad que el sistema migrado deberá enfrentar. Por tanto, es necesario validar si los modelos seleccionados o propuestos, de los proveedores seleccionados en el paso anterior, pueden o no cumplir con los requisitos de seguridad del LIS. Si no, el riesgo que no está cubierto por el requisito de seguridad no cumplido debe ser aceptado o un cambio en la arquitectura destino debe ser recomendada, proporcionando una lista de modelos aceptables que cumplen con los requisitos de seguridad.

II-B3. Actividad 3: Análisis: La actividad de análisis es en la que definimos los requisitos de seguridad a implementar e identificamos el conjunto de servicios de seguridad contratables a proveedores específicos de seguridad como servicio (SecaaS) que se integrarán en nuestra aplicación una vez migrada a la nube. También se identificarán otros controles tales como las cláusulas estándar del SLA que afectan a cuestiones de seguridad y también puede que volvamos a validar si los proveedores de servicio en caso de que algún proveedor concreto no pueda cumplir dentro de su marco contractual con los requisitos fundamentales de seguridad definidos.

- A3.1 Análisis de requisitos de seguridad en la nube

El modelo SMiLe2Cloud actualizado, proveedores cloud validados, modelos de servicio y despliegue son usados para derivar un conjunto de requisitos de seguridad con la cual el sistema diseñado debe cumplir con el nuevo entorno. Los requisitos serán un subconjunto de requisitos del LIS que el LIS ten?a y con los requisitos que se incluyeron en el desempe?o del análisis DAFO.

- A3.2 Asociación de los requisitos de seguridad con los elementos de SMiLe

Los artefactos obtenidos a partir de la tarea anterior deben ser utilizados para desarrollar un mapeo entre los requisitos de seguridad del LIS y una especificación formal de los requisitos de seguridad con la que el sistema destino debe cumplir para estar seguro de acuerdo con la especificación de la nueva arquitectura.

- A3.3 Análisis de los acuerdos estándar de nivel de servicio

Una vez que los requisitos de seguridad se han identificado y definido formalmente, es necesario seguir analizando el SLA estándar definido por los proveedores de la nube en busca de

problemas de seguridad, políticas de seguridad, elementos de seguridad que pueden ser medidos, etc.

- A3.4 Análisis de servicios de seguridad

La última tarea de esta actividad se ocupa de los servicios de seguridad actuales que son ofrecidos por los proveedores de servicios de seguridad. Una vez más, esto puede implicar el análisis de SLA de estos proveedores y mapear algunas cláusulas del SLA en requisitos de las actividades anteriores.

II-B4. Actividad 4: Diseño: En la actividad de diseño se definen los componentes propiamente dichos que forman el núcleo de la arquitectura de seguridad del sistema (cláusulas, controles personalizados, protocolos, etc.), y no sólo se define el diseño, sino que también se define la forma en la que deben ser validados y se planifican las actividades que serán necesarias en la migración real de la seguridad del sistema heredado.

- A4.1 Diseñar la arquitectura de seguridad básica para la nube

En esta tarea, se toma la especificación de los requisitos de seguridad y las cláusulas del SLA identificados en los pasos anteriores, junto con la lista de los anteriores servicios de seguridad cloud y se desarrolla la arquitectura de seguridad básica en términos de controles que se pueden ser integrados para cumplir con los requisitos de seguridad.

- A4.2 Diseñar los acuerdos personalizados de nivel de servicio

Siempre que sea posible, SLA (ya sea SLA general o SecSLA) debe ser personalizado para satisfacer las necesidades específicas del cliente. La mayoría de los analistas cloud aconsejan que los contratos de servicio se adapten a las necesidades del cliente. En la práctica, esto sólo será un motivo de preocupación para los grandes clientes que pueden negociar contratos lucrativos. Por otra parte, es evidente que no todos los proveedores de servicios permitirán la personalización de los servicios y/o cláusulas hasta el grado deseado.

- A4.3 Validar la arquitectura de seguridad específica de la nube

Una vez que la arquitectura de seguridad ha sido obtenida, y antes que la migración actual comience, tiene lugar la validación de la arquitectura. Esta validación involucra una revisión formal del diseño que hemos propuesto (ya sea SLA o controles personalizados). Después de esta validación, la aplicabilidad y viabilidad técnica de la arquitectura debería ser aclarada; es decir, todos los controles que se implementen a través de SLA deberían ser elegibles o dentro del ámbito SLA de los proveedores seleccionados y la responsabilidad de entregar el control siempre debe estar clara (es decir, cuando usamos dos proveedores de servicios, debemos asegurarnos que no hay ninguna posibilidad de que los contratos deleguen mutuamente la responsabilidad del control de seguridad). Como alternativa, los controles deben poder aplicarse como controles personalizados en el modelo seleccionado (es decir, en PaaS, el acceso está disponible para definir usuarios y otorgar permisos en una base de datos).

- A4.4 Planificar la estrategia de migración

Finalmente, la última tarea de la actividad de diseño es desarrollar un plan relativo a cómo la seguridad del LIS será implementada con recursos, horarios, logros, etc.

II-C. Actividad 5: Migración

Finalmente, la propia migración tiene lugar y es necesario contratar en la realidad los servicios y firmar los acuerdos de nivel de servicio y desarrollar los elementos de seguridad personalizados e implantarlos y configurarlos para dejar todos los controles de seguridad en condiciones de operación habitual.

- A5.1 Contratar servicios de seguridad

En este punto tiene lugar la formalización del contrato. Este contrato puede ser un acuerdo de nivel de servicio con un proveedor de servicios de seguridad en la nube o pueden ser las cláusulas específicas de seguridad que se definen en los contratos con proveedores de servicios IaaS, PaaS o Saas.

- A5.2 Desarrollar controles de seguridad a medida

Si nuestra arquitectura define controles de seguridad personalizados, ha llegado el momento de desarrollarlos. Por ejemplo, si hemos definido que nuestro sistema tendrá una pieza de software que controlará los perfiles de usuario en una base de datos ofrecida por un proveedor de PaaS que no incorpora un sistema de roles internamente en la propia base de datos, será necesario desarrollar la pieza de software que realice la gestión del roles e integrarla en nuestras aplicaciones y programas que desarrollan elementos funcionales; también será necesario en este punto hacer las pruebas unitarias de software de los controles de seguridad a medida.

- A5.3 Configurar controles de seguridad

Para los controles de seguridad personalizados definidos, contratados y/o implantados de forma personalizada en los pasos anteriores, normalmente es necesario realizar una función de despliegue en el sistema final. Además, si los controles necesitan algún tipo de configuración, en este punto deberán ser configurados y afinado su funcionamiento.

II-D. Actividad 6: Evaluación

Una vez que todo el proceso ha concluido y el sistema heredado ha sido movido a la nube de forma segura, es el momento de verificar y validar el sistema y los controles de seguridad.

- A6.1 Verificar seguridad del sistema cloud

En actividades anteriores (durante el análisis y el diseño) algunos de los artefactos de salida eran entradas en la parte del proceso y del modelo de seguridad que trata con las cuestiones de pruebas, verificación y certificación de la seguridad.

- A6.2 Validar seguridad del sistema cloud

Técnicamente, la validación es la actividad formal que hace que un sistema sea válido para el responsable de las cuestiones de seguridad de las tecnologías de la información: el administrador de la seguridad. La tarea consiste en revisar las evidencias obtenidas en la actividad anterior y en producir un documento que establece que la gestión de la seguridad está de acuerdo con la seguridad de los sistemas heredados (LIS) migrados a la nube de acuerdo con los requisitos especificados.

II-E. Actividad 7: Mejora

Dado que nuestro proceso tiene vocación de mejora continua (se trata de un ciclo de Deming) no finaliza con la validación real del sistema en funcionamiento. Periódicamente, el responsable de la seguridad del sistema heredado deberá reunir nuevas evidencias que permitan asegurar que el sistema está permanentemente configurado según los requisitos y parámetros de seguridad definidos y que permita renovar la validación. También estudiará mejoras que afecte al análisis DAFO, al análisis de seguridad en la nube o incluso a la lista de servicios en la nube que pueden ser considerados en las anteriores tareas.

- A7.1 Estudiar mejoras

La nube es un entorno cambiante. Algunos de los problemas que ahora están siendo objeto de estudio por parte de la mayor parte de los expertos, hace un par de años ni siquiera se conocían. En un par de años, puede que haya servicios completamente nuevos que ayuden a fortalecer la seguridad de un sistema heredado migrado a la nube. Además, dado que al mover un sistema a la nube, delegamos la responsabilidad sobre la aplicación de algunos controles, es necesaria y aconsejable que se vigilen los niveles y métricas definidos para asegurar su cumplimiento.

- A7.2 Renegociar cuestiones de seguridad

Finalmente, hemos definido una actividad que permita renegociar con los proveedores de servicios y proveedores de seguridad las incidencias de seguridad. Esta negociación es diferente de la que supone la renegociación de nuevos servicios.

III. Conclusión

En este artículo hemos presentado un proceso que permite la migración de la seguridad o la migración segura a la nube de un sistema de información heredado. Comenzamos en el punto en el que el sistema ha sido objeto de un proceso de ingeniería inversa y tenemos disponibles una serie de modelos KDM que definen la parte funcional del sistema heredado. Desde este punto, ofrecemos una serie de actividades que permitirán evolucionar estas especificaciones en formato KDM en una arquitectura de seguridad para el sistema heredado y desde allí en un sistema objetivo migrado a la nube en forma segura; actualmente estamos desarrollando técnicas y plantillas para automatizar parcialmente el proceso de entrega de una arquitectura segura y para mapear la arquitectura de seguridad deseada en un modelo que de forma específica trate las cuestiones específicas de la nube como las amenazas específicas que la nube presenta, los requisitos de seguridad específicos para la nube, los controles específicos relacionados con la nube (ya sean en su forma de seguridad como servicio o como controles personalizados); todo ello con la intención de que una aplicación heredad que sea migrada a la nube cumpla estándares de seguridad en la nube tales como la matriz de controles de la CSA. Nuestro trabajo futuro se enfocará en un refinamiento del propio proceso y en el desarrollo de herramientas y patrones que permitan de forma semiautomática asistir al analista de seguridad en las actividades de obtención del modelo de seguridad del sistema heredado y la derivación del modelo de seguridad del sistema migrado a la nube a partir de aquél. La aplicación real de migración de un sistema heredado al cloud se definirá y ejecutará siguiendo SMiLe2Cloud.

Agradecimientos

Esta investigación es parte de los siguientes proyectos: GEODAS (TIN2012-37493-C03-01) y SIGMA-CC (TIN2012-36904) financiados por el "Ministerio de Economía y Competitividad y Fondo Europeo de Desarrollo Regional FEDER", España.

Referencias

[1] W. Jansen, T. Grance, "Guidelines on Security and Privacy in Cloud Computing".*NIST SP - 800-144*, 2011.

[2] J.R.V. Winkler, in "Securing the Cloud. Cloud Computing Security. Techniques and Tactics", B. Meine, Editor. Syngress, Elsevier. p. 25. 2011.

[3] M. Tobin, and B. Bass. "Federal Application Modernization Road Trip: Express Lane or Detour Ahead?". MeriTalk. 2011.

[4] V. Kundra, "Federal Cloud Computing Strategy", U.S.C.I. Office, Editor. 2011.

[5] W. Zhang, A. J.Berre, D. Roman, and H. Aage Huru. "Migrating Legacy Applications to the Service Cloud", in OOPSLA 2009, Towards Best Practices in Cloud Computing. 2009.

[6] S. Frey and W. Hasselbrind. "Model-Based Migration of Legacy Software Systems into the Cloud: The CloudMIG Approach", in 12 Workshop on Software-Reengineering of the GI-SRE. 2010.

[7] H. Zhou, H. Yang, and A. Hugill. "An Ontology-Based Approach to Reengineering Enterprise Software for Cloud Computing", in IEEE 34th Annual Computer Software and Applications Conference. 2010. Seoul, Korea. p. 383-388.

[8] Q.H. Vu and R. Asal, "Legacy Application Migration to the Cloud: Practicability and Methodology", in IEEE Eighth World Congress on Services. 2012.

[9] D.G. Rosado, R. Gómez, D. Mellado, and E. Fernández-Medina, "Security Analysis in the Migration to Cloud Environments". Future Internet, 2012. 4(2): p. 469-487.

[10] R. Gomez, D.G. Rosado, D. Mellado, and E. Fernández-Medina, "Security Criteria in Deciding on Migration of Systems to the Cloud", in 9th International Workshop on Security in Information Systems. 2012: Wroclaw, Poland. p. 93-100.

[11] L. Marquez Alcañiz, D.G. Rosado, D. Mellado, and E. Fernández-Medina, "Security in legacy migration to the cloud: a systematic mapping study", in 11th International Workshop on Security in Information Systems. 2014: Lisbon, Portugal. p. 26-37.

[12] R. Seacord, D. Plakosh, and G. Lewis, "Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices". 1st ed. 2003: Addison Wesley.

[13] KDM, "Knowledge Discovery Meta-Model", Version 1.3. OMG specification formal 2010-12-12. 2011.

[14] CSA, "Security Guidance for Critical Areas of Focus in Cloud Computing V3.0". 2011.

Virtual TPM for a secure cloud: fallacy or reality?

Jordi Cucurull
Research and Security department,
Scytl Secure Electronic Voting
Email: jordi.cucurull@scytl.com

Sandra Guasch
Research and Security department,
Scytl Secure Electronic Voting
Email: sandra.guasch@scytl.com

***Abstract*—The cloud technology has dramatically increased the virtualisation usage during the last years. Nevertheless, the virtualisation has also imposed some challenges on the security of the cloud. A remarkable case is in the usage of cryptographic hardware such as the Trusted Platform Module (TPM).**

A TPM is a device, physically attached to a server, that provides several cryptographic functionalities to offer a foundation of trust for the running software. Unfortunately, the virtualisation of the TPM to bring its security properties to virtual environments is not direct due to its design and security constraints.

During the last years several proposals have been presented to solve the virtualisation of the TPM. Nevertheless, the virtualisation systems have not started to adopt them until very recently. This paper reviews three existing implementations of virtual TPM in the Xen and QEMU virtualisation solutions. The main contribution of the paper is an analysis of these solutions from a security perspective.

***Palabras clave*—cloud, security, TPM, vTPM, virtualisation, XEN, QEMU**

I. Introduction

The cloud technology has dramatically increased the usage of virtualisation during the last years. Virtualisation has detached the software applications from the physical machines where they are hosted. This allows a sensible use of resources, since the same infrastructure can be shared by many applications and the number of running servers can be adapted to the load. Nevertheless, the virtualisation has also introduced new challenges to the security of the clouds. A remarkable case is the one where the machines were using secure hardware to ensure the integrity of the infrastructure, such a Trusted Platform Module (TPM).

A TPM [1] is a device, physically attached to a server, that provides different cryptographic functionalities to facilitate the creation of a foundation of trust of the software installed in the server. Unfortunately, the TPM was not designed with virtualisation in mind, hence its virtualisation is not direct and it implies several security considerations.

Since Berger et al. [2] presented their work about virtualisation of the TPM, a few other proposals have appeared [3], [4], [5], [6]. Nevertheless, existing virtualisation software did not seem to adopt any of these solutions until very recently.

The main purpose of this paper is the analysis of three identified implementations of virtual TPM (vTPM) for the Xen and QEMU virtualisation solutions. The paper is organised in seven sections. Section II presents the technologies related to the implementations analysed, Section III describes the vTPM solution for Xen, Section IV describes two vTPM solutions for QEMU, Section V analyses the security of the solutions presented, Section VI discusses the security findings, and Section VII presents the final conclusion of the paper.

II. Technologies

This section describes the technologies that might be involved in a virtualised Trusted Platform Module solution.

A. Platform virtualisation

Platform virtualisation is the practise of emulating one or more physical hosts, or parts of them, within an actual physical host. The software that creates and manages the virtual guests (emulated hosts) within the host machine (physical host) is the hypervisor. Two types of virtualisation can be distinguished:

- **Full virtualisation**: The physical host is fully emulated. The operating system of the virtual guest does not realise is running on an emulated device and it does not require any modification. This solution can be based only on software or leverage specific hardware virtualisation extensions of the CPU [7], which provide different performance.
- **Paravirtualisation**: The physical host is emulated with selected modifications of its architecture to enhance the scalability, performance and simplicity of the solution [8]. The operating system of the virtual guest has to be adapted to work with the emulated host.

In this article we selected the Xen [9], [10] and QEMU [11], [12] virtualisation systems. Xen is a system that supports full and paravirtualised x86 guests. Xen maps the virtual guests as domains. There is a privileged domain, called Dom0, and user domains, called DomU. Additionally, some of the functionality of Dom0 was disaggregated in Stub Domains [13] for security and scalability purposes. QEMU is a fully virtualised system that can emulate different architectures. As opposed to Xen that manages the whole host machine, QEMU is a standalone application within the host machine.

B. Trusted Platform Module

A Trusted Platform Module (TPM) [1] is a device, physically attached to a server and with a standard interface called TPM-TIS [14], that provides different cryptographic functionalities in the host, e.g. to ensure the integrity of the platform. A TPM includes a Root of Trust for Storage (RTS) for external secure key storage, non volatile protected storage (NVRAM), facilities to digitally sign data and the Platform

Configuration Registers (PCR) to store measurements of the system done by the TPM.

A TPM has at least 16 PCR registers, which are initialised to a known value when the machine is rebooted. The values of these registers cannot be arbitrarily set. Instead, they are modified by an operation called extension that performs a hash, a SHA1 in TPM 1.2 [1], of the previous value of the register and the piece of data to measure. The signed values of the PCR registers can be retrieved from the TPM by issuing the TPM Quote operation.

TPM is based on Public Key Infrastructure (PKI). The TPM has a special key, the Endorsement Key (EK), that is created by the TPM manufacturer and that can include a certificate issued by it. When the TPM is initialised by the user, in the process of taking the TPM ownership, the Storage Root Key (SRK) is generated. This key is the root of the hierarchy of keys that will be subsequently generated and used in the TPM. Finally, the Attestation Identity Keys (AIK) are used as an alias of the EK for signing information produced by the TPM, e.g. the PCR register values issued after the TPM Quote operation.

The TPM is mainly used to create a foundation of trust of the software installed in the host where the device is present. This is performed through a process called Static Root of Trust for Measurement (S-RTM) [15]. This process performs a chain of measurements, starting when the host platform is reset, of the components and configuration data involved in the system boot. Each component measures the next component before passing the control to it, forming what is called a Chain of Trust (CoT). The CoT, at least, involves the BIOS, the boot loader and the operating system kernel. The resulting measurements after a system boot, must be always the same unless the boot components are modified.

The combination of the TPM Quote operation and the S-RTM process, allows the remote attestation [16] of the host. An external attester can request a TPM Quote of the PCRs, and compare the obtained values with a baseline of the PCR values of the system generated when it was in a trusted state.

C. Virtual TPM

Virtualisation is currently an extended practise, but the TPM was not designed for virtualised systems. The security offered by a TPM is based on the principle of a trusted piece of hardware to create the foundation of trust in a given host. The implementation of a virtual TPM (vTPM) for a virtualised environment, to provide equivalent security than a physical TPM (pTPM), requires a special care with: protection of the vTPM secrets, link between the vTPMs and the virtual guests, extension of the CoT from the host machine to the virtual guests and key management. Several works analyse and proposes solutions to the virtualisation of the TPM [2], [17] and to its integration in virtualised systems [3], [4], [5], [6]. Nevertheless, it is not until recently that implementations have started to come up and be integrated in well-known virtualised environments (see Sections III and IV).

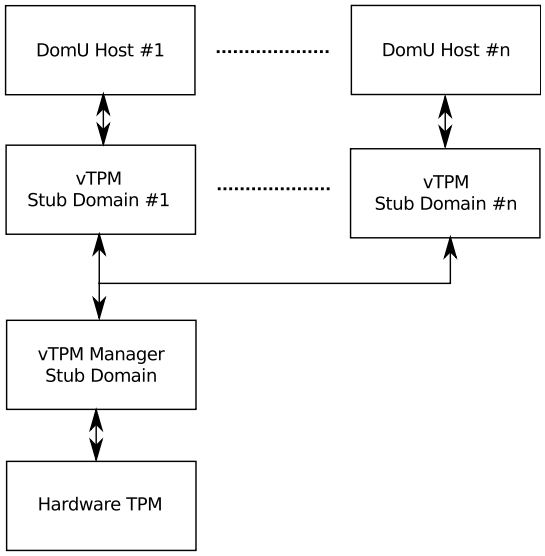


Figure 1. Structure of vTPM service in Xen

III. Virtual TPM in Xen

Xen 4.3 implements the service of virtual TPM (vTPM) only for paravirtualised guests.The service is designed as a set of secure separate stub domains (see Figure 1) managed by the system hypervisor, each of them running a mini-os [18] and its dedicated functionality. Each virtual guest has a software emulated TPM, based on the TPM Emulator [19], running in a vTPM stub domain. And there is a vTPM Manager stub domain that coordinates and links the vTPMs with the physical TPM (pTPM). The most relevant characteristics of the virtualised TPM implementation of Xen are:

- **Non transparent vTPM**: A custom kernel module driver (tpmfront) must be installed in each virtual guest. The module provides the standard TPM interface (`/dev/tpm`) to the applications of the virtual guest, i.e. they can use the vTPM as if it was a pTPM. The custom kernel module driver is not integrated in the current Linux kernels and it is not easy to find. In addition, the driver is not available for non Linux based operating systems.
- **vTPM's secrets bound to pTPM**: The secrets of the vTPM are encrypted with AES-256 and stored in disk. The symmetric key is bound to an storage RSA key of 2048 bits. The RSA key is generated by the pTPM and can only be used by it.
- **Configurable TPM ownership and SRK authentication**: The passwords used to access the pTPM and the SRK are configurable. These passwords must be provided at the time of loading the vTPM Manager stub domain.
- **Passthrough of certain Physical TPM registers**: The administrator of the vTPM stub domain can configure certain PCR registers of the vTPM to adopt the values of the same registers of the pTPM.
- **Extension of CoT from the host machine to the virtual guests**: If the "pv-grub" external bootloader is used to boot the virtual guest, the guest kernel is measured

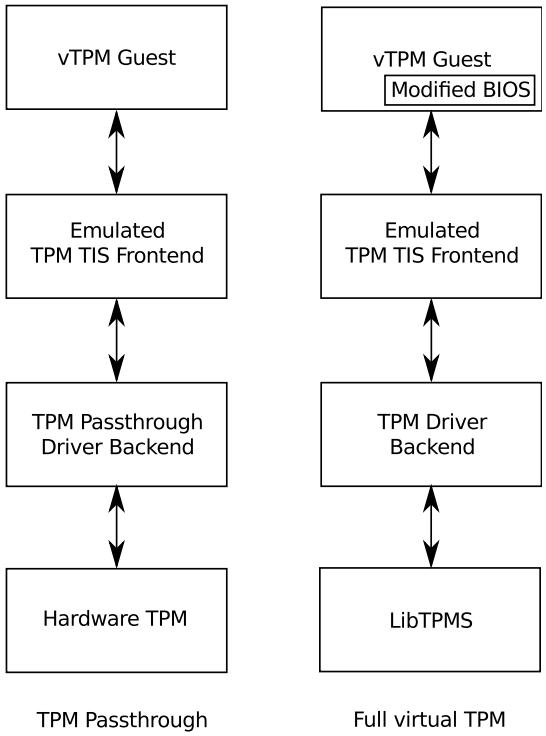


Figure 2. Structure of vTPM services in QEMU

in PCR #4 and the boot command line and initrd are measured in PCR #5 of the vTPM. Hence, the integrity of the guests can be ensured if the "pv-grub" bootloader, the hypervisor and other components that support the guests are trusted.

- **Migration of vTPM not supported**: All virtual TPMs are bound to a specific pTPM. Hence, guests with virtual TPMs cannot be migrated to another physical server.

IV. Virtual TPM in QEMU

QEMU supports virtual TPMs (vTPM) for guests from its version 1.5. Nevertheless, the only officially supported vTPM is based on TPM passthrough. This means that the TPM offered to the virtual guests is the actual physical TPM (pTPM) of the physical host. In addition, there is an implementation of the vTPM that is not officially integrated into QEMU that provides full vTPMs.

A. TPM passthrough

The TPM passthrough, as previously mentioned, provides a vTPM to the virtual guests which is a direct mapping with the pTPM of the physical machine. The service is designed as a backend driver for the pTPM that communicates with an emulated TPM TIS frontend (see Figure 2). The most relevant characteristics of this implementation are:

- **Transparent vTPM offered to guest host**: The virtual guest sees the vTPM as if it was a pTPM. No special kernel drivers are needed.
- **Passthrough of all the physical TPM registers**: The vTPM is a direct mapping of the pTPM. The PCR values, NVRAM area and keys of the vTPM are the same of the pTPM. Hence, all the measurements performed by the physical host are reflected to the vTPM.
- **Only one virtual guest can be provided with vTPM**: The reason is the one to one mapping between the vTPM and the pTPM. The registers and the NVRAM of the pTPM cannot be multiplexed to support multiple vTPMs.
- **Migration of vTPM is not supported**: The pTPM registers and NVRAM cannot be extracted from the TPM and imported into another pTPM. Hence, migration cannot be supported.

B. Full virtual TPMs

The full virtual TPMs approach, as previously mentioned, provides a complete vTPM implementation to the virtual guests that is totally detached from a pTPM. The service is designed as a software TPM backend implementation linked with the external library libTPMS. This library provides TPM emulation. On the guest side there is an emulated TPM TIS frontend (see Figure 2) and a modified open source BIOS, based on SeaBIOS [20], to support the vTPM. The most relevant characteristics of this implementation are:

- **Transparent vTPM offered to guest host**: The service of vTPM is based on full TPM emulation. Since the TPM TIS interface is emulated, no modifications have to be performed to the guest operating system.
- **vTPM's secrets stored into QEMU image**: The secrets of the vTPM are stored within an image file. The secrets are not encrypted by default, however QEMU allows the use of encrypted images, e.g. QCOW2 [21] can provide AES-128 encryption.
- **No pTPM required**: Since the vTPMs are fully emulated and not bound to a pTPM, this solution does not require the presence of a pTPM in the system.
- **Modified BIOS with vTPM and SRTM support**: A set of patches [1] to be applied to SeaBIOS are provided. The patches add vTPM support and implement the Static Root of Trust for Measurement (SRTM), i.e. the code that takes care of the first measurements right after the machine is powered on.
- **Migration ready**: The migration of the vTPM is not implemented, but it would not be difficult to integrate because the vTPM is not strongly linked to a pTPM.

V. Security considerations

There are four aspects of the vTPMs in virtualised environments that define their level of security regarding a pTPM: protection of the vTPM secrets, link between the vTPMs and the virtual guests, extension of the CoT from the host machine to the virtual guests and key hierarchies and management. This section analyses these four aspects for the vTPM implementations presented (see also Table I).

[1] See e-mail with patches of Stefan Berger "[PATCH V3 0/8] Add TPM support to SeaBIOS" of April 2011 in SeaBIOS mail list

Table I
COMPARISON OF VTPMS

	Xen	QEMU TPM Passthrough	QEMU Full vTPM
Approach	Multiple vTPMs	pTPM passthrough	Multiple vTPMs
Multiple virtual guests	Yes	No	Yes
Transparent vTPM	No	Yes	Yes
vTPM secrets	Digital envelope linked to pTPM (AES-CBC 256 bits and RSA 2048 bits)	pTPM	Image (allows AES-CBC 128 bits)
Link of vTPM and virtual guest	Weak	Static	Strong (state) / Weak (secrets)
Physical to virtual Chain of Trust	No, but includes external bootloader that measures kernel	No	No, but includes external BIOS with SRTM
Key hierarchies	Independent	Keys of pTPM	Independent
VM Migration	No	No	Yes
Orientation	Production	Development and testing	Production
Implementated in	Xen Project 4.3 and above	QEMU 1.5 and above	Experimental patches for development

A. Protection of the vTPM secrets

A TPM has data that must be kept secret and safe from manipulation, e.g. the Endorsement Key (EK) or the data contained in the NVRAM area.

The vTPM secrets in Xen are bound to the pTPM through a digital envelope. The data of the envelope is ciphered with AES-CBC symmetric encryption with a 256 bits key generated using the pTPM TRNG [22]. The symmetric key is protected with a public key of the pTPM. Hence the vTPM secrets can only be recovered accessing the pTPM. It is announced that in the future there will be the possibility to seal the symmetric key, i.e. the current protection linked to the PCR values of the pTPM. In that case if the hypervisor or Dom0 critical elements are corrupted, due to a change of the PCR values, the vTPM secrets will not be available.

In QEMU TPM Passthrough the vTPM secrets and registers are literally protected by the pTPM. This has the advantage of the hardware-based security offered by the pTPM, but it also means that anybody with access to the pTPM has access to the vTPM secrets.

In QEMU Full vTPM the secrets are kept in a dedicated image file without any protection mechanism implemented. Nevertheless, it is possible to leverage the security offered by the specific type of image used. Currently only QCOW2 offers privacy, in this case password based encryption with AES-CBC and 128 bits key. Nevertheless, the password is limited to 16 alphanumeric characters, hence its security level is limited to 105 bits. No authenticated encryption [23] nor other integrity-preserving mechanisms are used, hence the secrets could be manipulated. In addition, the system is not mature enough and it was failing when a QCOW2 encrypted image was used.

B. Link between vTPMs and virtual guests

The link between vTPMs and virtual guests must be protected. Otherwise, a virtual guest could be provided with a different, and probably manipulated, vTPM with measurements that may not correspond to the guest.

In Xen, the vTPM is completely independent of the virtual guest, including their lifecycles, and they run in different domains. The link between the vTPM stub domains and the virtual guest domains is not robust neither authenticated. Hence any vTPM domain can be linked to any virtual guest domain. In addition, it is possible to pause a virtual guest domain and replace its vTPM domain with the one of another guest, i.e. completely replacing PCR registers and non volatile data. This allows a corrupt admnistrator, or attacker with equivalent privileges, in Dom0, to manipulate the vTPM virtual guests association.

In QEMU TPM Passthrough the association between vTPM and virtual guest is static, since there is only one possible vTPM that is mapped to the pTPM. Despite this increases the security of the vTPM secrets in front of attackers without privileged rights, the vTPM lifecycle and state are mapped to the pTPM. Hence, anyone with access to the pTPM or to the host node can manipulate the measurements shown in the vTPM, e.g. by directly accessing the pTPM, enabling another virtual guest with access to it or rebooting the virtual guest (on reboot of the virtual guest the vTPM values are not initialised since the lifecycle of the vTPM are linked to the physical machine).

In QEMU Full vTPM, the vTPM is implemented and managed by the same instance of the hypervisor that manages the virtual guest. Hence the association with the vTPM and virtual guest lifecycle is implicit, i.e. there is no possibility to manipulate the PCR registers. Nevertheless, there is no strong link between the image file that contains the vTPM secrets and its virtual guest.

C. Chain of Trust extension to the virtual guests

In a virtualised TPM solution, the security offered by the vTPMs depends on the underlying host machine. It is desirable to create a Chain of Trust (CoT) in this host and link it to the individual CoTs created in each virtual guest. The verification of the CoT extension requires access to the measurements of both pTPM and vTPM to evaluate.

In Xen, the "pv-grub" guest bootloader allows the extension of the CoT from the host machine to the virtual guests. In detail, the bootloader measures, in the vTPM, the kernel, initrd and command line used to boot the guest. Additionally, the bootloader can be measured in one of the registers of the pTPM and, this register, be selected to be shown as one of the vTPM PCRs. In this case, both CoT of the host machine and virtual guest would be linked. Nevertheless, the authors of the solution discourage the direct usage of the pTPM in the host machine. The reason is that if the pTPM drivers and software stack are installed in Dom0, the administrators of the system have easy access to the pTPM and, as a consequence, to the key used to encrypt the data of the vTPMs. This prevents the usage of the pTPM for measuring and checking the integrity of Dom0 and extending the CoT. Nevertheless, this will be solved in the new-coming releases of the solution.

In QEMU TPM Passthrough, it is not possible to extend the CoT of the physical host to the virtual guest since the PCR registers of the pTPM and vTPM are the same, the lifecycle of the vTPM is linked to the one of the pTPM and the bootloader of the guest image cannot be measured by QEMU. In this solution there is no clear border between the security of the physical host and the security of the virtual guest.

In QEMU Full vTPM, the modified BIOS provides support to link the CoT of both the node machine and the virtual guests. The modified SeaBIOS implements the S-RTM process, which allows to create a CoT within the the virtual guest. If the BIOS of the virtual guests, the hypervisor and other software managing the system is measured in the host machine, the link between the host machine and virtual guests CoT can be created.

In all the cases where the CoT extension would be possible, the system is vulnerable to malicious administrators replacing the bootloaders during runtime, virtual guest BIOS, or any other software involved in the virtual guest management.

D. Key hierarchies and management

All TPMs have at least an EK and, after its ownership is taken, a SRK which is the root for its key hierarchy.

In the full vTPM implementations in Xen and QEMU the keys are completely independent of the ones present in the pTPM. Despite this implies a loosely coupled key hierarchy with a pTPM, in practise will facilitate the migration of the vTPMs when this becomes ready in the future. In Xen the EK is automatically generated the first time the vTPM is initiated, while in QEMU the EK has to be explicitly generated by the user issuing a specific command from within the virtual guest. In the QEMU TPM Passthrough implementation, the keys used in the vTPM are the same used in the pTPM.

Additional options exist, when the key hierarchy of a vTPM is generated [2], in order to provide keys that may become certified by a certificate authority. Nevertheless, the current vTPM implementations still do not offer them.

Regarding the key generation, in Xen and QEMU TPM Passthrough the pTPM TRNG is used as random number generator. While the QEMU Full vTPM implementation uses a random number generator provided by the OpenSSL library.

VI. Discussion

Given the security considerations detailed in Section V, it can be stated that the security currently provided by the existing vTPMs implementations is not equivalent to the security of a pTPM. In all the cases, the security of the virtual guests depend on the administrators of the machine hosts. Nevertheless, the fully virtualised vTPMs of Xen and QEMU set the bases for a near future usage of this technology.

In Xen, if there is a malicious administrator in the physical host, the security offered by the vTPM of the virtual guests cannot be guaranteed. This is something known by the authors of this implementation [2]. As they state, the solution is to create a domain building component measured by the pTPM during boot. This component should have a static library with the critical domains to build. This component should enforce the creation and destruction of these domains as well as the correct pairing of vTPM domains and guests. We believe that an administrator should not be allowed to log into the machine without modifying the measurements of the TPM, e.g. the login could add a measurement to the TPM of each user that logs into the system. This could be used as a tamper-proof mechanism. Hence the physical machine would become a kind of sealed box.

In QEMU, the difference with Xen is that there is no hypervisor that controls the whole virtualised system. In this case, for a maximum security, a software manager of the virtual guests should be installed in the physical host. The manager should ensure the image file that contains the vTPM secrets and the modified BIOS were correctly paired to the correct virtual guest to ensure its integrity. This manager could be measured as part of the physical machine CoT. In this case it would also be possible to extend the CoT of the physical host to the virtual guest, assuming the patched SeaBIOS is in place and a secure bootloader is installed in the virtual guest. Since the BIOS code used in QEMU is explicitly provided when the guest is started, the tool that manages the guest machines could ensure its integrity. As in Xen, the physical host could generate measurements in the TPM for each user logged, hence it would become as a sealed box in the sense that nobody can log to perform system changes without being detected.

VII. Conclusions

In this article we have analysed two virtualisation solutions with three currently available virtual TPM approaches. The purpose of this analysis was to determine if there where available virtualised TPM solutions and the level of security offered by them.

After the presented analysis we found two implementations that offer complete TPM virtualisation for Xen and QEMU. The implementation in Xen still has not reached a level of

[2] See "Questions about the usage of the vTPM implemented in Xen 4.3" in February 2014 in the xen-devel mailing list.

security comparable to a non virtualised solution, but their developers are pushing hard for it. In addition, the solution is integrated within the Xen official releases. The implementation in QEMU offers less security than the one in Xen, e.g. to store the secrets of the vTPM, and its integration with QEMU is not officially supported due to restrictions of the project for including code that has dependencies with external libraries (in this case because of the libTPMS).

Given the development activity seen, it is expected the improvement of the security and availability of the virtualised TPM solutions soon. In addition, the virtualised systems will integrate other technologies that enhance the trust with their hypervisor, e.g. the support of the IntelTXT technology [24] that simplifies the foundation of trust for the hypervisors in virtualised systems in conjunction with the TPM.

Acknowledgements

We want to thank the developers of the Xen and QEMU vTPM solutions for the information given through the different development forums. This work has been co-funded by the project Trusted Cloud IPT-2011-1166-430000 of the Ministry of Economy and Competitiveness (MINECO) and the European Fund for Regional Development (FEDER)".

References

[1] TCG, "TPM Main Specification Level 2 Version 1.2," March 2011.
[2] S. Berger, R. Cáceres, K. A. Goldman, R. Perez, R. Sailer, and L. van Doorn, "vTPM: Virtualizing the Trusted Platform Module," in *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, ser. USENIX-SS'06. Berkeley, CA, USA: USENIX Association, 2006.
[3] S. Berger, R. Cáceres, D. Pendarakis, R. Sailer, E. Valdez, R. Perez, W. Schildhauer, and D. Srinivasan, "TVDc: Managing security in the trusted virtual datacenter," *SIGOPS Oper. Syst. Rev.*, vol. 42, no. 1, pp. 40–47, Jan. 2008.
[4] B. Danev, R. J. Masti, G. O. Karame, and S. Capkun, "Enabling secure VM-vTPM migration in private clouds," in *Proceedings of the 27th Annual Computer Security Applications Conference*, ser. ACSAC '11. New York, NY, USA: ACM, 2011, pp. 187–196.
[5] D. Liu, J. Lee, J. Jang, S. Nepal, and J. Zic, "A cloud architecture of virtual trusted platform modules," in *Embedded and Ubiquitous Computing (EUC), 2010 IEEE/IFIP 8th International Conference on*, Dec 2010, pp. 804–811.
[6] D. Wallom, M. Turilli, G. Taylor, N. Hargreaves, A. Martin, A. Raun, and A. McMoran, "myTrustedCloud: Trusted cloud infrastructure for security-critical computation and data managment," in *Cloud Computing Technology and Science (CloudCom), 2011 IEEE Third International Conference on*. IEEE, 2011, pp. 247–254.
[7] K. Adams and O. Agesen, "A comparison of software and hardware techniques for x86 virtualization," *SIGARCH Comput. Archit. News*, vol. 34, no. 5, pp. 2–13, Oct. 2006.
[8] A. Whitaker, M. Shaw, and S. D. Gribble, "Denali: Lightweight virtual machines for distributed and networked applications," in *In Proceedings of the USENIX Annual Technical Conference*, 2002.
[9] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," *ACM SIGOPS Operating Systems Review*, vol. 37, no. 5, pp. 164–177, 2003.
[10] "Xen Project," http://www.xenproject.org.
[11] F. Bellard, "QEMU, a fast and portable dynamic translator," in *USENIX Annual Technical Conference, FREENIX Track*, 2005, pp. 41–46.
[12] "QEMU," http://wiki.qemu.org.
[13] S. Thibault, "Stub domains: A step towards dom0 disaggregation," Xen Summit, 2008, http://blog.xen.org/index.php/2008/08/28/xen-33-feature-stub-domains/.
[14] TCG, "PC Client Work Group PC Client Specific TPM Interface Specification (TIS), Version 1.3," March 2013.
[15] TCG, "PC Client Work Group Specific Implementation Specification for Conventional Bios Specification, Version 1.2," February 2012.
[16] G. Coker, J. Guttman, P. Loscocco, A. Herzog, J. Millen, B. O'Hanlon, J. Ramsdell, A. Segall, J. Sheehy, and B. Sniffen, "Principles of remote attestation," *International Journal of Information Security*, vol. 10, no. 2, pp. 63–81, 2011.
[17] F. Stumpf and C. Eckert, "Enhancing trusted platform modules with hardware-based virtualization techniques," in *Emerging Security Information, Systems and Technologies, 2008. SECURWARE'08. Second International Conference on*. IEEE, 2008, pp. 1–9.
[18] S. Popuri, "A tour of the mini-os kernel," http://www.cs.uic.edu/~spopuri/minios.html.
[19] M. Strasser and H. Stamer, "A software-based trusted platform module emulator," in *Trusted Computing - Challenges and Applications*, ser. Lecture Notes in Computer Science, P. Lipp, A.-R. Sadeghi, and K.-M. Koch, Eds. Springer Berlin Heidelberg, 2008, vol. 4968, pp. 33–47.
[20] "SeaBIOS," http://www.seabios.org/SeaBIOS.
[21] M. McLoughlin, "The QCOW2 image format," September 2008, https://people.gnome.org/~markmc/qcow-image-format.html.
[22] A. Suciu and T. Carean, "Benchmarking the true random number generator of TPM chips," *CoRR*, vol. abs/1008.2223, 2010.
[23] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," in *Advances in Cryptology - ASIACRYPT 2000*, ser. Lecture Notes in Computer Science, T. Okamoto, Ed. Springer Berlin Heidelberg, 2000, vol. 1976, pp. 531–545.
[24] Intel Corporation, "Intel TXT software development guide," 2014, http://download.intel.com/technology/security/downloads/315168.pdf.

Information System for Supporting Location-based Routing Protocols

Gerard Garcia, Sergi Robles, Adrià Sánchez, Carlos Borrego
Departamento de Ingeniería de la Información y de las Comunicaciones
Universidad Autónoma de Barcelona
Email: {*ggarcia, sergi.robles, adria.sanchez, cborrego*}@deic.uab.cat

***Abstract*—This article presents an information system for location-based routing protocols that does not compromise the privacy of the involved nodes. This information system provides a representational model of the most frequented locations of a node, this most frequented zone is called its habitat, and a protocol to compare these habitats among nodes given a target location of interest. Then, the protocol can determine which of the neighbors of a node is nearer or farther from this target location and provides this information to the underlying routing protocol. As it is designed for DTNs, the protocol does not require a trusted third party, instead, it implements a secure multi-party computation based on homomorphic encryption. The protocol is demonstrated to be secure against passive and active adversaries.**

***Index Terms*—Secure multi-party computation, Delay and Disruption Tolerant Network, DTN routing protocol.**

I. Introduction

The Delay and Disruption Tolerant Network (DTN) [7] architecture emerged from the research for developing an interplanetary network. This research was focused in solving the technical difficulties found in out of space networking, p.e. frequent disconnections or slow network links. But as these characteristics could also be found in some scenarios on the Earth, the DTN term was coined in order to include them.

One of the most important challenges that DTNs have to face because of its characteristics is how to perform the routing of messages [16]. The differences between DTNs and traditional networks, such as the lack of an end-to-end circuit between the source and the destination or the fact that nodes can not have a global knowledge of the network due to their disconnected nature, make the routing protocols used in traditional networks ineffective. To overcome this, some routing protocols for DTNs generate metrics that model the behavior of the nodes in the network. Then, with the information provided by the model, the routing protocol can make substantiated decisions on how to forward the messages in order to try to maximize the delivery rate.

This paper presents an information system support for location-based routing protocols. This information system is composed of a representation model of the most frequented locations of a node and a protocol for determining, given n nodes, which of their habitats is nearer or farther of a target location.

This information support would allow a location-based routing protocol the maximization of the delivery rate of the messages by forwarding them to the nodes nearer to the destination location of the given message. Or it would also allow a routing protocol to forward the messages by paths that try to avoid specific zones by sending the messages to the nodes farther of these zones.

To implement the protocol would seem logical to think that the coordinates of the target location could be directly compared to the habitats of the nodes. But the privacy of the involved nodes must be taken into account, since this nodes could identify physical persons and revealing its location could be a threat and an invasion of their privacy. Therefore, the habitats comparisons are treated as a secure multi-party computation [13].

There are several solutions for performing a secure multi-party computation. The ones originally proposed by [9] and [15], and extended by many others, make use of a combinatorial circuit for representing the required computation. Parties execute then a short protocol for every gate of the circuit. The advantage of these approaches is that they are general methods, but the problem is that the protocol depends on the size of the circuit. Therefore, for complex computations these protocols can be inefficient. Other approaches, like the ones proposed by [17] or [2], design specific protocols based on, for example, homomorphic encryption or 1-out-of-N oblivious transfers, to solve specific problems. These solutions are more efficient, but are limited to the solving of these specific problems. The proposal in this paper uses a specific secure multi-party computation based on homomorphic encryption for efficiency reasons.

II. Habitat

This section first describes what is and how is represented an habitat. Then, it shows how the habitats of two nodes con be compared given a target location. And finally, shows how is calculated the distance between two points.

A. Description

The habitat of a node represents its most frequented locations and it is represented by a dynamically created ellipse from the historic of movement of the node. How is the ellipse created is not contemplated in this article. An ellipse can be defined as

Definition 1. *The set of points such that the sum of the distances to two fixed points, the foci, is constant. This distance defines the radius of the ellipse.*

Hence, an habitat is defined with two foci points $F1 : (f1_x, f1_y)$ and $F2 : (f2_x, f2_y)$ and a radius r.

B. Comparison

When comparing the habitats of two nodes, three different situation may be found:

1) The target location is outside the two habitats. In this case the one nearer to the target is preferred.
2) The target location is inside the two habitats. In this case the node with the smallest habitat is preferred, as it is more probable that the node pass through this location earlier.
3) The target location is inside one of the habitats but outside the other. In this case the preferred node is the one with the target location inside its habitat.

To solve each one of the previous situations, it is necessary to solve the following three problems:

1) How to calculate the distance from an habitat to a target location.
2) Given two habitats, how to determine which one is smaller.
3) How to determine if the target location is inside an habitat.

1) Distance from a target location to an habitat: It is necessary to calculate the distance from the target location defined by a point $P : (x, y)$ to an habitat H, determined by the ellipse with foci points $F1 : (f1_x, f1_y)$ and $F2 : (f2_x, f2_y)$ and radius r.

First, it is defined the point $X : (a, b)$ as the nearest point of the habitat H to the point P, so it would need to comply the next equation

$$|a - f1_x| + |b - f1_y| + |a - f2_x| + |b - f2_y| = r \quad (1)$$

Then, it is defined the function distance as follows

$$d(X, P) = |a - x| + |y - b| \quad (2)$$

and it is minimized while restricted by equation 1, for example with the method of Lagrange multipliers, to get the point X. Finally, to obtain the distance is applied the function distance 2 with the point X and P.

2) Which habitat is smaller: To know which habitat is smaller, are compared the radius of the two habitats. Given two habitats H_1 and H_2, with radius r_1 and r_2 respectively, the one with the smallest radius is the smallest habitat

$$\begin{aligned} r_1 < r_2 &\implies H_1 \\ r_2 < r_1 &\implies H_2 \end{aligned} \quad (3)$$

3) Point inside an habitat: This problem can be resolved as the first one. If the distance obtained is negative or 0, then the point is inside the habitat.

C. Manhattan Geometry

To simplify the previous calculations it is used the definition of distance that Manhattan Geometry [5] provides. In Manhattan Geometry the function distance of two points is defined as the sum of the absolute differences of their Cartesian coordinates. Formally

$$d(p, q) = \sum_{i=1}^{n} |p_i - q_i| \quad (4)$$

where p and q are two points. This way it is simpler to calculate the distance while still can be compared, as it maintains the proportions. This simplification will be of interest when calculating the distance in the secure multi-party computation as the operations that can be performed will be limited by the use of homomorphic encryption,

III. Cryptographic protocol for habitat comparisons

In this section first will be described how the problems for comparing two habitats, described in section II, are solved such that the privacy of the involved nodes remains unaffected. Then, it will be showed how the protocol works for determining, given n nodes, which one, or ones, are nearer or farther to the target location. The protocol will implement a secure multi-party computation based on homomorphic encryption to perform the calculations needed to compare the habitats of the nodes.

A. Homomorphic encryption

An encryption scheme is considered homomorphic if given the set of plain-texts $\mathcal{M}$, the set of the cypher-texts $\mathcal{C}$ and the encryption function $\mathcal{E}$, it satisfies

$$\forall m_1, m_2 \in \mathcal{M}, \mathcal{E}(m_1 \odot_{\mathcal{M}} m_2)) \leftarrow \mathcal{E}(m_1) \odot_{\mathcal{C}} \mathcal{E}(m_2) \quad (5)$$

for some operators $\odot_{\mathcal{M}}$ in $\mathcal{M}$ and $\odot_{\mathcal{C}}$ in $\mathcal{C}$ [8]. If the encryption scheme only satisfies this property for one operation, e.g. multiplication or addition, it is considered partially homomorphic.

For this protocol are used the homomorphic properties of the cryptosystem proposed by P. Paillier in [11], known as the Paillier cryptosystem. This cryptosystem is additively homomorphic, computationally efficient and it allows the multiplication of cyphered-texts by unencrypted constants without the need of decrypting the operands. Therefore, the set of operations cryptographically protected that can be performed are: sum, subtraction and multiplication of an encrypted value by a non-encrypted constant.

B. Secure comparison

The use of homomorphic encryption limits the operations that can be performed over the encrypted operands, therefore, the previous comparison process needs to be adapted to overcome these limitations.

The problem appears when the distance from an habitat to a target location is calculated. To calculate the distance it is necessary to first find the nearest point of the habitat to

the target location and then use the found point to calculate the distance, as described in II-B1. But if the operands are encrypted under homomorphic encryption it is not possible to calculate their absolute value, therefore this operation needs to be disposed.

To do so, it is determined where is the target location situated in relation of the habitat to make sure that all the subtractions encode an absolute value.

If the space is divided into 9 regions, it can be determined in which region is the target location calculating the maximum and minimum values of the foci points $(Fx_{min}, Fx_{max}, Fy_{min}, Fy_{max})$ and comparing them with the coordinates of P. Once it is known in which region is situated the target location, the equations 1 and 2 can be redefined without the need of calculating absolute values and then minimized for each of the cases.

Note that the corner regions define b in terms of a. This is the line where X is located, but not any point of this line is the nearest to P. To know exactly where is the nearest point each one of these regions is divided in two subregions. To do so is calculated the left end LE and the right end RE of the habitat and then compared with the target location. These two ends are calculated with the following equations

$$\begin{aligned} LE &= Fx_{max} + Fx_{min} + Fy_{max} - Fy_{min} - r \\ RE &= Fx_{max} + Fx_{min} - Fy_{max} + Fy_{min} + r \end{aligned} \tag{6}$$

If the point is located between LE and RE, X and P share the x coordinate, so $a = x$, otherwise, $b = Fy_{max}$ in the two superior corners and $b = Fy_{min}$ in the other two.

C. Protocol

To describe the protocol it is imagined an scenario where A has multiple neighbors but only A can see all of them. Given that this situation can be quite common, the protocol has been designed such that A will coordinate all the messages. Therefore, the protocol is based in letting A have the information that other nodes need to compare between them so A can distribute it between the other nodes as it requires. Obviously, as the privacy needs to be preserved, this information is encrypted and only accessible by the node referred.

The protocol is divided in two phases, which are represented in figure 1. After the first phase, A has compared its own habitat to the habitats of its neighbors. And during the second phase the surviving nodes (the nodes that still satisfy the requisites of A) are compared among them, under the commands of A, until A decides that it has enough information for its purposes.

The neighbor discovery is conducted with the transmission and detection of periodically transmitted beacons. When a node detects a beacon, if it has messages to forward, it sends a beacon asking all the receiving nodes to announce its presence so it can detect all the neighbor nodes at once. After a time t_1 has elapsed, the protocol continues.

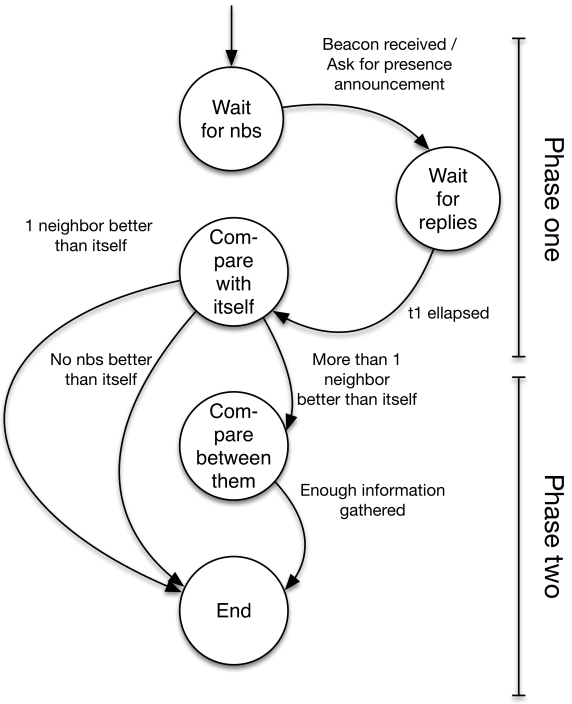


Figure 1. State diagram of the proposed protocol

1) Phase one: Node A compare its habitat with the habitat of another node following the exchange of messages described in figure 2. A is the node that coordinates the protocol and B any other neighbor.

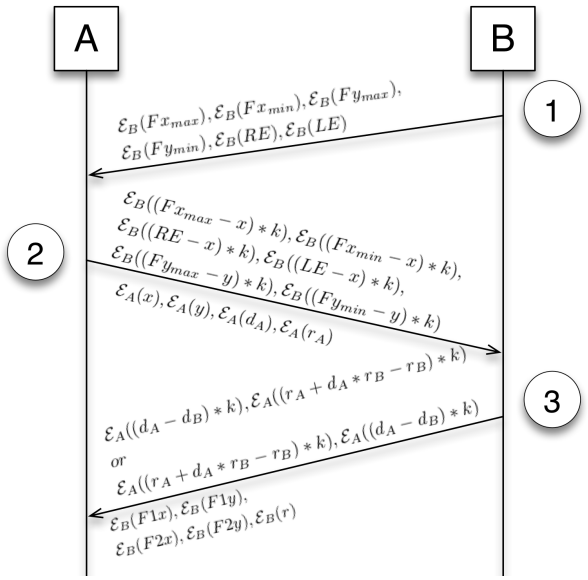


Figure 2. Phase one: Exchange of messages to compare a given habitat and a target location known by A with the habitat of other nodes

(1) Node B sends the limits of its habitat together with its presence beacon.

- Node A subtracts to the limits the coordinates of the target location so node B can determine in which region is situated the target location respect its habitat. Note that all the results are obfuscated, by multiplying them by k, so node B can only know the situation of the target location respect its habitat and not the exact position of it.

② Node A transmits the subtracted limits together with the target location $P : (x, y)$, its distance d_A (which node A has calculated without any restriction, as he knows the target location an its habitat), and its radius r_A.

- Node B calculates $\mathcal{E}_{\mathcal{A}}(d_B)$ with the supplied target locations, then, it creates the tuple described in the figure. This is done this way to do not let node A know if the distances are equal (or 0) as it will not distinguish between the comparison of the distance and the comparison of the radius.

③ Node B transmits the results to node A, and as node A needs to have the required information so other nodes can compare among them, node B includes with the result its own habitat ($F1$, $F2$ and r), encrypted with its own key.

- To interpret the result node A checks the sign of the two values of the tuple, if the values are 0 or positive, node B is considered to be nearer than A to the target location.
- With the habitat and knowing in which region is the target location, node A can calculate the distance of that node to it, although it is encrypted and unaccessible for itself. But as node A does not know the region where is the target location situated, it calculates the distance for each of the regions, so it ends having 13 different distances, one for each region and subregion.

Once the protocol finishes, node A knows if node B is nearer or farther to the target location than itself, but it has not learned anything about the habitat of node A. And on the other hand, node B has not learned the target location, only the situation of its habitat respect to this target location and has been able to compare its habitat with the habitat of node A without learning it.

If no nodes satisfy the requisites of node A, or only one node satisfies them, the protocol ends. Otherwise, the protocol continues with phase two.

2) Phase two: Phase two starts with node A knowing which of its nodes satisfy its requisites, and having the distance of the habitat to the destination target location of these nodes, although the distance is encrypted and node A does not know which of the 13 distances is the correct one.

At this point, node A has to determine which comparisons need to be done to reach a decision. As a method to compare the habitat of any two nodes is provided, any filtering or sorting algorithm can be used. For example, if node A only wants to know which node is the nearest to the target location, A could just make all nodes compare randomly while discards the ones that lose a comparison until only one node is left. Or if A wants to sort all its neighbors from nearer to farther, it can perform a Quicksort ordering. This decision is beyond the scope of this article.

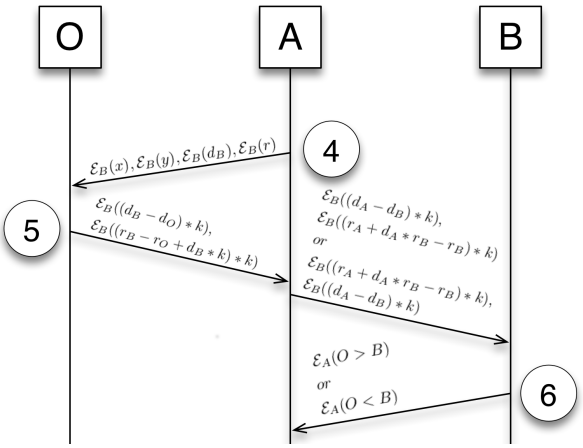


Figure 3. Phase two: Exchange of messages to compare the habitat of B with the habitat of other nodes

To perform this comparisons node A starts the exchange of messages shown in figure 3. Being node B the node to be compared an O the others nodes to compare with B:

④ Node A sends to the other nodes the information that they need to compare themselves with node B: the distances of node B to the destination target location d_B (13 of them, one for each region and subregion), the target location $P : (x, y)$ and the radius r_B of the habitat of node B. All this values are encrypted with the key of B.

- This is the same information that they received from node A in the first step of the protocol, with the difference that it is encrypted by node B and that they do not know which one is the correct distance, so they need to compare for each one.
  With this information they calculate their distance to the target location again, but encrypted for node B, and they subtract it to each of the d_B provided.

⑤ The next step requires opening the result of the comparison. Only node B can open the results, therefore, there are two options:

- If they can see each other, they send the result directly to node B,
- If they can not see each other, they send it to node A so node A can forward it to node B.

Once node B has the result, it opens the correct one.

⑥ Finally, node B transmits the interpreted results to node A.

This comparisons can be done in both ways, therefore, if node A does not completely trust node B it can ask node B to compare itself with the other nodes and only accept the results that match.

D. Performance

The information transfered by the protocol can be calculated with the following formula, b is the size of each value, m is the

number of messages ready to be forwared by a given node and c is the number of comparisons performed in the phase two (which number will depend of the chosen sorting algorithm):

$$12 * b + 11 * b * m + (17 * b * m + 2 * b) * c$$

The size of each value will depend on the key used to encrypt them, being them twice the size of the key.

It is safe to assume that most DTNs have a window time of at least a few seconds, for example [4] and [12], and a bandwidth in the order of, at least, the hundreds of kilobytes per second, thus, the overhead of the protocol would not impact on the performance of the network.

Regarding computation performance, it will depend on the hardware of the nodes, but as the operations to encrypt and decrypt of the Paillier cryptosystem have low complexity, it is feasible to implement the protocol in any moderately powerful hardware.

Finally, note that this information system is designed to complement other routing protocols, therefore it should be used when its use is expected to improve the delivery ratio of the messages.

IV. Security Discussion

The global security of the proposed protocol is not determined by the individual behavior of the nodes, therefore, the security rests in three main points: the Paillier cryptosystem, the key management and the control of the information indirectly disclosed. The security of the first two points can be quickly determined: The Paillier cryptosystem is proved to be robust and secure [11] and the security of the key management is responsibility of the node as the keys are never transmitted during the protocol. More interesting is the analysis of the information disclosed.

To analyze the information disclosed by the execution of the protocol, the framework defined in [6] for the analysis of the security of multi-party computations will be used as starting point. This framework describes different adversary models and gives a generalization of the concept of ideal process, already proposed by other authors in [10] and [3].

In this framework a protocol is considered to securely perform a given task if executing the real protocol amounts to "emulating" the ideal process for that task. In the ideal process there is an incorruptible trusted party who receives the inputs of all the parties, locally computes the desired outputs and transmits them to the required parties. To define what is "emulating", first is necessary to formalize the output of performing a given task.

The output is formalized as the information that the task explicitly outputs in addition of what can be inferred. In other words, the information the task outputs once it has successfully finished and the information that can be deduced from the process of performing the task.

Now, emulating a task is performing it in such a way that its output is exactly the same as the ideal task. Thus, all parties will learn identical information from both the real protocol and the ideal process.

The adversaries covered in this analysis are classified in passive and active adversaries. Passive adversaries, also called, semi-honest, only gather information and do not modify the behavior of the parties. On the other hand active adversaries, also called "Byzantine", modify the outputs of the function so they can corrupt other parties to get more information.

To simplify the study of the security of the proposed protocol, it will be divided in two subtask. The subtask of comparing the habitats of two nodes when one of them knows the target location, and the subtask of comparing a given habitat and a target location to n other nodes. The first subtask corresponds to the first phase of the protocol, and the second subtask to the second phase. These phases are described in section III. Also, it will be distinguished between the security of the nodes (the privacy of their habitats) and the security of the target location (as it can be the location of a node).

The analysis will be performed as follows. For each subtask and adversary, it will be described the output of performing the subtask in the ideal process and then it will be compared with the output of performing the same subtask in the real protocol.

A. One-to-one habitat comparison subtask

In the step one of the protocol it is performed a comparison between the habitats of two nodes, one of them knowing the target destination. Then, the task of this subtask is the determination, between these two habitats, of which one is better suited to forward a message to this destination. The nodes will be called A and B, being A the one with the message to forward.

To perform the ideal process of this task, A would send its habitat and the target location P to the trusted third party using a secure channel and B its own habitat with another secure channel. The output that the trusted third party would transmit would be the communication to the A of which of the habitats is considered better. Therefore, A would not learn any additional information about the habitat of B and B would not learn anything about the habitat of A nor the target location.

Now, will be show how this subtask behaves against passive and security adversaries.

1) Passive adversaries: The execution of the ideal process in presence of passive adversaries would not lead to any opportunity for them to gather additional information. Otherwise, the real protocol reveals to B the region where the target location P is located regarding its habitat.

The revelation of the region is the only difference between the ideal process and the real protocol, but if it points to a node this is not a threat to its privacy as its location can be hidden by breaking the relation between the target location and the node, p.e. with the technique used in [14]. Therefore it can be stated that this part of the protocol does not compromise the security of the forwarding nodes nor the nodes to which the target location can refer in presence of passive adversaries.

2) Active adversaries: In the ideal process, the active adversaries can only modify the inputs sent to the trusted third party, on the other hand, in the real protocol, both A and B

can modify the values in the intermediate steps of the protocol to try to corrupt the other node.

If the modified values were the values operated with the encrypted data, the consequences would be the same than modifying the original inputs. Thus, this modifications would not affect the security of the protocol as it has been defined because this attacks could be also performed in the ideal process. In addition, no other alterations can be done that would give any advantage to the adversary as all the values are encrypted and any modification to these values would produce random uncontrolled outputs when decrypted.

Hence, it is possible to state that this part of the protocol is also secure against active adversaries as it does not compromise the security of the nodes not of the target location.

B. One-to-N habitat comparison subtask

The other subtask is the comparison of a given habitat among other n nodes. The task is determining which one or which ones, depending the sorting or filtering process used, fulfill the requirements of the initiator of the protocol. It starts at the last step of the first phase of the protocol, when the node gives, together with the result of the comparison, its encrypted habitat. This subtask performs the part of the protocol used in the previous subtask but with different parties, then, only the differences from the previous subtask need to be discussed.

To perform this subtask in the ideal model, all the nodes would send its habitat to the trusted third party, and the node A, in addition, would send the target location P. Now, the trusted third party would perform the required calculations and would send the results of these comparisons to the node A.

The main difference from the previous subtask is that node A is allowed to have the information regarding the habitat of another node, but it is not able to access to it as it is encrypted. Then, once node A has calculated the distance of the target node using its habitat information, the protocol behaves in the same way than in the first subtask and no additional information is disclosed.

Therefore, as if node A has the information of another node does not affect the security of the protocol it is possible to assume that this subtask is also secure against passive and active adversaries. Thus, the whole protocol is considered secure against active and passive adversaries.

V. Conclusion

In this article we have proposed a support information system for location-based routing protocols that does not compromise the privacy of the involved nodes. This system allows the nodes to use geographical information to make routing decisions. With the proposed system, it is possible to forward the messages in such a way that they take a specific path. For example, the messages can be sent as directly as possible to their destination, avoiding specific areas...

All the required calculations are securely performed on the involved nodes while protected by the homomorphic Paillier cryptosystem. Then, since no trusted third parties are needed, this protocol is suitable for DTNs.

These calculations can be considered a specific case of a secure multi-party computation. Hence, the security of the protocol has been analyzed from this point of view. This analysis has concluded that the protocol is secure against passive and active adversaries, including collusion attacks.

As future work, it would be interesting to implement the proposed information system, e.g. in the aDTN platform currently developed by the SeNDA research group [1]. This platform allows the exchange of messages following the store-carry-process-and-forward paradigm proposed in [4] which allows the messages to provide their own routing code. Then, it would be of interest to develop several routing algorithms that make use of this information system and compare the performance of these routing algorithms to other routing protocols.

Acknowledgment

This work was partly supported by grant TIN2010-15764 from the Ministerio de Ciencia e Innovación of Spain and by grant 2014SGR-691 from Generalitat de Catalunya.

References

[1] Adtn implementation. https://senda.uab.cat/wiki/aDTN.
[2] Mikhail J Atallah and Wenliang Du. Secure multi-party computational geometry. In *Algorithms and Data Structures*, pages 165–179. Springer, 2001.
[3] Donald Beaver. Foundations of secure interactive computing. In *Advances in Cryptology—CRYPTO'91*, pages 377–391. Springer, 1992.
[4] Carlos Borrego, Sergio Castillo, and Sergi Robles. Striving for sensing: Taming your mobile code to share a robot sensor network. *Information Sciences*, 2014. In press.
[5] Donald R Byrkit. Taxicab geometry—a non-euclidean geometry of lattice points. *The Mathematics Teacher*, pages 418–422, 1971.
[6] Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, 2000.
[7] Kevin Fall and Stephen Farrell. Dtn: an architectural retrospective. *Selected Areas in Communications, IEEE Journal on*, 26(5):828–836, 2008.
[8] Caroline Fontaine and Fabien Galand. A survey of homomorphic encryption for nonspecialists. *EURASIP Journal on Information Security*, 2007, 2007.
[9] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 218–229. ACM, 1987.
[10] Silvio Micali and Phillip Rogaway. Secure computation. In *Advances in Cryptology—CRYPTO'91*, pages 392–404. Springer, 1992.
[11] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in cryptology—EUROCRYPT'99*, pages 223–238. Springer, 1999.
[12] Nagasai Panchakarla, Jörg Ott, and Gesner Junior. Delay-tolerant adaptive real-time communication: a case study for voice. *Proceedings of ExtremeCom'12*, 2012.
[13] Manoj Prabhakaran and Amit Sahai. *Secure Multi-Party Computation*, volume 10. IOS Press, 2013.
[14] Xiaoxin Wu and Bharat Bhargava. Ao2p: ad hoc on-demand position-based private routing protocol. *Mobile Computing, IEEE Transactions on*, 4(4):335–348, 2005.
[15] Andrew Chi-Chih Yao. Protocols for secure computations. In *FOCS*, volume 82, pages 160–164, 1982.
[16] Zhensheng Zhang. Routing in intermittently connected mobile ad hoc networks and delay tolerant networks: overview and challenges. *Communications Surveys & Tutorials, IEEE*, 8(1):24–37, 2006.
[17] Ge Zhong, Ian Goldberg, and Urs Hengartner. Louis, lester and pierre: Three protocols for location privacy. In *Privacy Enhancing Technologies*, pages 62–76. Springer, 2007.

$SoNeUCON_{ADM}$: the administrative model for $SoNeUCON_{ABC}$ usage control model

Lorena González-Manzano
Univ. Carlos III de Madrid
lgmanzan@inf.uc3m.es

Ana I. González-Tablas
Univ. Carlos III de Madrid
aigonzal@inf.uc3m.es

José M. de Fuentes
Univ. Carlos III de Madrid
jfuentes@inf.uc3m.es

Arturo Ribagorda
Univ. Carlos III de Madrid
arturo@inf.uc3m.es

***Abstract*—The popularity of Web Based Social Networks (WBSNs) encourages their enhancement. Many WBSN data is considered personal data and access control management plays a key role in this regard. The point is not only to manage access control but to determine how administration should be performed. Based on $SoNeUCON_{ABC}$, an expressive usage control model that allows fine-grained access control management, this paper presents $SoNeUCON_{ADM}$, the complementary administrative model. Based on a pair of related and popular administrative models, the evaluation proves the completeness of $SoNeUCON_{ADM}$.**

***Index Terms*—Administrative access control model, Web Based Social Network, revocation, delegation.**

I. Introduction

In Web Based Social Networks (WBSNs) users upload huge quantity of data, some of them personal data, which are in many cases let out of control. Controlling and carefully managing all WBSNs data is a demanding and challenging necessity. At a primary step, $SoNeUCON_{ABC}$, an expressive usage control model that allows fine-grained access control management along the whole usage process is proposed in [1]. However, access control models have to describe the way administration is performed and then, the identification and specification of administrative tasks for $SoNeUCON_{ABC}$ is the following step.

Coming back to the 90's, given the maturity of the Role Based Access Control Model (RBAC) proposed by R. Shadu *et al.* [2], its attached administrative model can be used as a precedent in the identification of administrative tasks [3]. In a nutshell, in RBAC, administrative permissions (analogous to rights) are exclusively applied to administrative roles and other permissions are applied to any other kind of roles. Then, administrative tasks base on the assignment of users to roles; the assignment of permissions to roles; and the assignment of roles to roles. The initial set of administrative tasks are summarized as follows:

- Who is the entity in charge of creating, updating and deleting access control preferences.
- Who is the entity in charge of associating preferences with data.
- How preferences are associated with data and data with data owners.

Furthermore, administrative issues also involve administrative rights management. Two types of rights are distinguished, namely, use and administrative rights. Use rights consist of operations performed with objects, e.g read right, and administrative rights correspond to operations performed over the right of objects, e.g. the right to give read right. The management of both types of rights is essential and delegation and revocation are remarkable operations in this regard. Delegation focuses on granting a right to a user, while revocation undoes the effects of delegation. In particular, *weak* and *strong revocation* are differentiated. The former refers to simply remove granted permissions and the latter refers to recursively revoke permissions from those to whom the grantee granted the permissions. Based on these rights and operations, the following administrative tasks are added to the previous ones:

- Who is the entity in charge of managing revocation.
- Who is the entity in charge of managing delegation.
- How weak and strong revocation is managed based on use rights and administrative rights.
- How delegation is managed based on use rights and administrative rights.

In the social networking field administration focuses on managing uploaded resources like photos or videos, specified identity data (namely personal profiles) and established access control policies. Thus, WBSN administrative tasks are equivalent to the ones above mentioned but considering that resources, identity data and policies are the elements at stake. As a result, this paper presents $SoNeUCON_{ADM}$, an administrative model for $SoNeUCON_{ABC}$. $SoNeUCON_{ADM}$ addresses all aforementioned tasks to promote a wider use of $SoNeUCON_{ABC}$.

This paper is structured as follows. Related work is described in Section II. Section III presents the background. Section IV introduces administrative features, particularly, tasks and rights. In Section V $SoNeUCON_{ADM}$ is described. The evaluation of the model is described in Section VI. Lastly, conclusions and future work is outlined in Section VII.

II. Related work

This Section presents the analysis of 21 proposals in the literature that address administrative issues in collaborative environments. Note that this study is not exclusively focused on WBSNs, but extended to collaborative environments due to a pair of reasons. On the one hand, WBSNs manage data which may be related to multiple users and then, they can be pointed out as collaborative systems. On the other hand, a

TABLE I
ADMINISTRATIVE FEATURES ANALYSIS

	Proposals	Administration	Delegation	Revocation
[25]	B. Carminati et al. (2011)	D		
[6]	A.C. Squicciarini et al. (2009)	D		√*
[24]	H. Zhang et al. (2012)	C		
[5]	M.R. Thompson et al. (2003)	D	√	√
[7]	A.C. Squicciarini et al. (2010)	D		√*
[8]	A. Ahmad et al. (2012)	D	√	√
[9]	Y. Jung et al. (2013)	D		√
[10]	Y. Ren et al. (2011)	D		
[11]	M. Prilla et al. (2006)	D		√
[12]	A. Imine et al. (2009)	D		√
[13]	M. Lorch et al. (2003)	D	√	√
[14]	H.F. Wedde et al. (2003)	D		
[15]	R. S. Shandu et al. (2010)	D	√	√
[16]	R. S. Shandu et al. (2011)	D	√	√
[17]	W.K. Edwards (1996)	C		
[18]	K. Sikkel et al. (1997)	D	√	√
[19]	Z.Y. Zhang et al. (2011)	D	√	√
[20]	R.K. Thomas (1997)	C		
[21]	E. Cohen et al. (2002)	D	√*	
[22]	V. Gligor et al. (2002)	D		√*
[23]	J. Jin et al. (2006)	D	√	

∗: mentioned but not managed

small amount of proposals focus on administrative issues in the specific context of WBSNs.

In general, 6 contributions fall in the WBSN category [4], [5], [6], [7], [8], [9], 3 proposals in document sharing [10], [11], [12], one proposal bases on grid environments [13] and the rest of them focus on other general collaborative systems [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24].

This analysis studied the administration type, namely, centralized C (a single entity decides who can get into the systems) or decentralized D (multiple entities decide who can get into the systems); and how delegation and revocation are managed. Table I presents results of the analysis. Symbol * means that a particular feature has been mentioned but not managed.

In what concerns the administration type, 18 approaches deal with D administration and just 3 proposals focus on C administration. As expected, administration tends to be decentralized because each WBSN user has to manage his owned data.

Concerning centralized administration, in [17] a central administrator manages roles and policies. Furthermore, the need of dynamism is highlighted and the change of user roles, at runtime, is an essential matter to deal with. Similarly, [20] proposes teams management. Teams are composed of users with the same role whose management is left to a general administrator. Likewise, in [24] groups are managed by a central authority in such a way that users are added to groups and rules, based on user attributes, time periods and resource usages, are applied to groups.

The majority of approaches base on decentralized administration, allowing users to individually manage their personal data. For instance, in [12], the administrators initiate the administration process by notifying updates to affected users who become involved in the administrative management process. By contrast, in [23], [5] users who want to become involved in a particular administrative process have to request it. Other proposals divide data, particularly documents, among users and they work over each owned piece of data [10]. A different solution are proposed by M.R. Thompson *et al.* [5] and A. Ahmad *et al* [8]. M.R. Thompson *et al's* work bases on certificates jointly signed by all users involved in the administrative process. However, A. Ahmad *et al* propose *transfer*, *multiplication* and *division* operations [8].

Delegation, associated with decentralized administration, is addressed in a total of 9 approaches. In collaborative environments several users have to cooperate to achieve a common goal. Then, delegating permissions breaks the power of a central administrative user by sharing administrative tasks among different parties. The most of approaches focus on permissions delegation [15], [16], [18], [13], [5], [24], [8], being the proposals of Z.Y. Zhang *et al.* and J. Jin [23] the only ones which propose role delegation [19] and E. Cohen *et al.*'s proposal which exclusively mentions the difficulty in managing delegation in organizational environments [21].

Related to revocation management, in multiple cases users may regret having granted a certain use or administrative right to a user. A total of 10 proposals provide mechanisms to deal with revocation and other 3 contributions mention the relevance of its management [22], [6], [7]. They focus on weak revocation in respect to rights [15], [16], [11], [13], [8] and group memberships [12] and on strong revocation regarding delegated rights [18], [19], [9] and certificates [5].

In sum, it is concluded that administration in collaborative environments tends to be decentralized. This is specially remarkable in WBSNs, where many users and data are managed. Besides, most of analysed approaches propose revocation and delegation mechanisms which helps to conclude the relevance of their management as part of the administration process.

III. BACKGROUND: $SoNeUCON_{ABC}$

$SoNeUCON_{ABC}$ is an expressive usage control model that manages six WBSN features, namely, common-contacts, clique, distance, multi-path, direction and flexible attributes [26], [27], [28], [29].

In general, $SoNeUCON_{ABC}$ is composed of seven elements: *Subjects* (S) together with *Subject attributes* ($ATT(S)$) refer to WBSN users and their attributes; *Objects* (O) together with *Object attributes* ($ATT(O)$) correspond to WBSN data and their attribute; and *Relationships* (RT) together with *Relationship attributes* ($ATT(RT)$) refer to the set of relations and attributes that exist between a pair of users, being direct relationships denoted as E and $ATT(E)$ their attached attributes; *Rights* (R) correspond to actions that can be performed over objects O; *Authorizations* (A) refer to rules to satisfy to grant a subject a right on an object; *Obligations* (B) correspond to requirements to satisfy before or while the usage process; and *Conditions* (C) refer to requirements to satisfy in regard to context features, eg. network availability.

In $SoNeUCON_{ABC}$, access control policies, denoted as ρ, consist of $\rho(\rho_s; \rho_o; \rho_{rt}; r; \partial_b; \partial_c)$. In particular, ρ_s, ρ_o and ρ_{rt} are predicates defined over subject, object and relationship attributes respectively. Besides, rights are denoted as r and obligations and conditions refer to ∂_b and ∂_c respectively.

In the following, an example of an access control policy is presented: *Access is granted to photos entitled "Party" to friends of a friend if they are under 30 years old or if they are under 25 years and have studied computer science.*

$\rho = (((age < 30) \vee ((age < 25) \wedge (studies = c.science)))$; $(title = party)$; $((((role = friend)$; $(role = friend))), \emptyset, \emptyset)$; $read$; $\emptyset$; $\emptyset)$

For more details of $SoNeUCON_{ABC}$ usage control model see [1].

IV. Towards administration

Prior to the description of how administration is performed in $SoNeUCON_{ADM}$, administrative tasks to address (Section IV-A) and the available rights to manage (Section IV-B) are detailed in the following Sections.

A. Administrative tasks

Administration involves multiple tasks (recall Section I) which can be classified in a couple of groups regarding tasks related to:

- *The identification of who is involved in administrative issues*. These tasks refer to who manages access control policies, who associates policies with resources and identity data and who manages revocation and delegation.
- *The definition of how administrative issues are performed*. These tasks correspond to how policies are associated with resources and identity data, how resources and identity data are associated with their owners and how revocation and delegation are managed.

B. Rights management

Two types of rights are differentiated, *use rights* and *administrative rights*. The former ones, which are referred in $SoNeUCON_{UCON}$ to as Rights (R), base on operations performed with objects such as read, and operations carried out over objects like tag, move or copy. By contrast, *administrative rights* (AR) refer to the management of elements involved in the access control decision process, along with delegation and revocation management.

V. $SoNeUCON_{ADM}$ definition

Users enrolled in a WBSN become owners of uploaded resources, established identity data (mainly profile data) and defined access control policies. Thus, $SoNeUCON_{ADM}$ is based on **ownership**, such that owned elements are managed by their owners. Specifically, administrative objects (AO) correspond to the elements involved in the access control decision process, namely, managed subjects (S), objects (O), direct relationships (E) and their respective attributes ($ATT(S)$, $ATT(O)$,$ATT(E)$) and access control policies (ACP).

In $SoNeUCON_{ADM}$, owners execute administrative rights AR over administrative objects AO and grant use rights R over objects O according to access control policies ACP (see Figure 1). In this regard, following Sections describe use rights R and administrative rights AR management (Section V-A and V-B respectively).

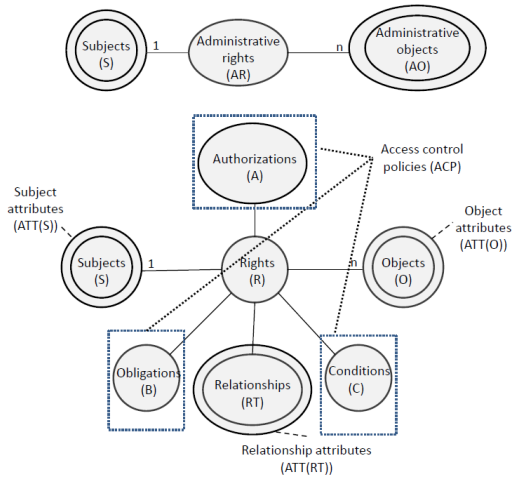


Fig. 1. $SoNeUCON_{ADM}$

A. Use rights management

Each owner specifies as many access control policies as desired and leaves them in a pool of policies to be evaluated when a request is received for executing some right over one of his owned objects. Contrary to other models, policies in ACP are not directly associated with data and its owner but to the owner exclusively. For instance, the policy "*grant read access to data entitled PARTY to users older than 20*" is created, associated with an owner and located in his pool of policies. Next, when an object of a particular owner is requested, all policies associated with him are evaluated, verifying authorizations (A), composed of subjects, objects and relationship attributes and the granted right ($ATT(S)$, $ATT(O)$, $ATT(RT)$ and r), obligations (∂_b) and conditions (∂_c). If there is a policy ρ_i within the set of policies defined by an owner (P_{ow_i}) that matches the request, the right r over the requested object o is granted to the requester s. Assuming that the expression $owner(\text{"}element\text{"})$ means being owner of "element", it is formally defined as:

(s, o, r) $granted \Leftarrow P_{ow_i}$= $\{\rho_i \in ACP / owner(\rho_i) = owner(o)\} \wedge$
$\exists \rho_i (A(ATT(S), ATT(O), ATT(RT), r); \partial_b; \partial_c) \in P_{ow_i} /$
$\rho_i (A(ATT(s), ATT(o), ATT(rt(owner(o), s)), r); \partial_b; \partial_c) = true$

B. Administrative rights management

This Section details the management of administrative objects (AO), revocation and delegation. In general, being owner of a particular administrative object ao grants administrative rights AR over it to manage the object and its attributes and to delegate and revoke use rights R and administrative rights AR over it. It is formally defined as:

$(s, ao, management)$ granted$\Leftarrow s = owner(ao)$
$(s, ao, delegation)$ granted$\Leftarrow s = owner(ao)$
$(s, ao, revocation)$ granted$\Leftarrow s = owner(ao)$

1) Administrative objects management: Administrative objects AO management consists of the creation, modification and deletion of any AO.

In terms of subjects S, they can create WBSN accounts, becoming owners of their profiles, uploaded data and established access control policies. Analogous, they can cancel their accounts whenever desired.

Objects O are other topic for discussion. In general, objects are stored in WBSN data bases, eg. Facebook. Nonetheless, in decentralized WBSNs, like Diaspora, each user chooses the host to store his data. Similar to WBSN accounts, objects have to be deleted when users want.

In regard to direct relationship E (as the indirect once are constructed through them [1]), WBSN users establish relationships with other users, as well as they update or remove them.

Concerning attributes, subject, objects and relationships attributes have to be considered ($ATT(S)$, $ATT(O)$ and $ATT(E)$ respectively). Firstly, $ATT(S)$ which basically refer to profile data, are linked to a WBSN account and they can be established by the account's owner, retrieved from an Identity Provider (IdP) where they where previously defined or obtained from personal devices like identity cards. Second, $ATT(O)$ attached to an object can be defined by its owner, as well as retrieved from the object's metadata. Nonetheless, if required, owners have to give permission to WBSNs to process metadata. Finally, what concerns with $ATT(E)$, they are considered identity data and then, they can be defined by owners or retrieved from IdPs.

On the other hand, access control policies can be also created, updated or deleted, at any time. In particular, all subjects with a WBSN account can manage access control policies ACP.

One last point is that the use of attributes, conditions and obligations in spite of being opened sets, depends on what every WBSN supports.

2) Delegation management: Delegations consist of granting permission to a certain user over a particular object temporary or permanent. The delegation of use rights R can be analogous to the establishment of access control policies. A right is granted to the requester over the requested object after satisfying an access control policy.

On the contrary, the delegation of AR requires the definition of the following function:

- DELEGATE(v_k,v_j,o_i,λ): It states that v_k gives a specific AR λ to v_j over o_i. λ refers to a partial or a complete delegation, the former to delegate some AR and the latter to delegate all AR. λ takes the value $*$ for a complete delegation and takes the value, e.g., AR-R to express that only the permission to grant use rights R is delegated. Note that this administrative model applies permanent delegation and the temporal one is left as a matter of future work.

In $SoNeUCON_{ADM}$ the delegation of AR compels the permanent delegation of all AR. Thus, the object over which the operation is executed, becomes property of the delegatee. The delegation operation should be enforced such as λ takes the value $*$, DELEGATE(v_k,v_j,o_i,$*$).

3) Revocation management: Revocation, contrary to delegation, removes the granted right over an object to a certain user. There are two types of revocation, weak and strong (Section I). Nonetheless, weak revocation of use rights R is the only $SoNeUCON_{ADM}$ manages since the delegation of AR is permanent and recursive delegation of use rights R are not applied.

$SoNeUCON_{ADM}$ manages revocation in terms of the update of attributes and access control policies, eg. if a photo entitled "Summer" is accessible to relatives, it would remain accessible to this set of people until the policy or the photo's title change. Indeed, it is extremely related to usage control and the application of *mutability* and *continuity* attributes. *Mutability* refers to the fact that attributes can be updated at any time. On the other hand, *continuity* refers to the enforcement of access control along the whole usage process. Both attributes are directly related to revocation because if initial conditions change along the usage process, access decisions have to be taken again [30] and they may cause the revocation of granted rights. Based on [31], revocation can be also divided between direct and indirect:

- *Direct revocation* can be enforced, at any time, by the owners of resources and identity data. Data owners may decide to revoke rights previously granted, updating or deleting an access control policy, as well as changing attributes. For instance, if the right to access a photo entitled "Classes" is granted to relationships with role "classmates", revocations can be caused by the update of the title of the photo or by the update of the role of a classmate relationship. Likewise, if the policy " Grant access to Friends to all photos" is updated to "Grant access to Friends to photos entitled Birthday", it may prevent requesters from getting requested rights in subsequent requests or while the usage process.
  In the revocation process, apart from the data owner, the Usage Reference Monitor is the entity at stake. This entity is composed of a Usage Decision Facility (UDF) and a Usage Enforcement Facility (UEF) which are always active [34] and they are applied in the usage control process. UDF identifies changes in attributes and UEF enforces access control accordingly. When policies are updated or attributes are changed, the UDF is informed about that. Afterwards, it informs the occurred event to the UEF and lastly, the UEF enforces the re-evaluation of policies.
- *Indirect revocation* is caused by uncontrolled situations. Particularly, it is performed when access control policy attributes expire or change. "Automatic" attributes updates, either subjects, objects or relationship attributes, can cause revocation of granting rights."Automatic' means that no users interactions are required. For instance, if the right to access a photo entitled "High-school" is granted to users under 18, revocations occur when requesters turn to 18 years old. Note that "automatic" updates are

TABLE II
ADMINISTRATIVE TASKS COMPARISON

Tasks	**SoNeUCON**_ADM	**UCON**_ABC [32], [33]	**RBAC** [3]
Entities identification			
Creating, updating and deleting access control preferences	Owners.	Owners.	Owners.
Associating preferences to data	Not required	-	Owners
Revocation management	Usage reference monitor and owners	-	Owners
Delegation management	Owners	-	Owners
Management procedures			
Association between preferences with data and data with data owners	Policies are exclusively associated to data owners concerning subjects, objects and relationships attributes.	Assertions associate subjects and objects	Permissions are associated with roles and and data and roles with data owners
Revocation management	Weak revocation is managed. Attributes and access control policies updates.	Weak revocation is managed. Time assigned to access control policies.	Weak and strong revocation are managed. Owners revoke users from roles according to their decisions.
Delegation management	Delegation of R and all AR is available. Owners establish access control policies and execute the delegation operation for all AR.	Delegation of R. Assertions associated with particular requesters.	Delegation of R and AR is available. Owners assigned users to roles to delegate.

specially related to attributes in which time is directly or indirectly involved.

The management is equivalent to *direct revocation* except for the fact that the UDF identifies updated attributes.

VI. EVALUATION

This Section presents the evaluation of $SoNeUCON_{ADM}$, the administrative model for $SoNeUCON_{ABC}$. It consists of comparing the proposed model with the most challenging and related administrative models, RBAC and $UCON_{ABC}$. $SoNeUCON_{ADM}$ is compared with RBAC administrative model, for being one of the most mature administrative models [35], [3], and with $UCON_{ABC}$ administrative capabilities, for being the model that lays the bases on the proposed one [32], [33].

Administrative tasks, identified in Section I, are depicted and compared in Table II, where symbol '-' implies that a particular task is not studied.

Concerning the association of data with preferences and data with data owners, $SoNeUCON_{ADM}$ only requires to associate preferences (access control policies) to data. Policies are mainly defined over subjects, objects and relationships attributes instead of being attached to specific objects. By contrast, $UCON_{ABC}$ and RBAC pose more restrictive and tedious tasks from the users point of view. In $UCON_{ABC}$ owners define assertions to associate subjects with objects, as well as to associate policies (composed of assertions) with objects [33]. However, in RBAC permissions are assigned to roles and to objects and then, roles are assigned to users.

Delegation is also managed in all compared models, being the $SoNeUCON_{ADM}$ proposal the most flexible one. In $SoNeUCON_{ADM}$ delegating R involves the establishment of access control policies according to subjects, objects and relationship attributes. Moreover, the delegation of all AR involves the execution of the operation DELEGATE to guarantee that, from the moment the operation is enforced, the delegated object becomes property of the delegatee without the possibility of undoing the operation. Conversely, delegation in $UCON_{ABC}$ is limited to R. It bases on specifying assertions associated with particular requesters which base on objects and subjects attributes [33]. On the other hand, RBAC delegates R and AR through the association of roles to users.

Revocation is another compared task. $SoNeUCON_{ADM}$ manages direct and indirect revocation. The former is performed by owners through the change of attributes and access control policies. On the contrary, indirect revocation is exclusively related to attributes updates, being particularly related to attributes involving time restrictions. Nonetheless, as this model only delegates R and all AR, just weak revocation is at stake. Similarly, $UCON_{ABC}$ manages weak revocation assigning time to access control policies. Moreover, though not described in the original model, Z. Zhang *et al.* proposed a general procedure to manage weak and strong revocation in $UCON_{ABC}$ [36]. On the other hand, RBAC provides functions to weakly and strongly revoke users from roles by removing the assignment of users to roles.

In the light of the proposed analysis, $SoNeUCON_{ADM}$ supports all tasks an administrative model should provide and thus, their completeness is pointed out. Indeed, $SoNeUCON_{ADM}$ has a significant advantage, that is, preferences (access control policies) are associated to users instead of to objects and the burden of managing at least as many policies as uploaded objects is avoided. Moreover, it is noticeable that $SoNeUCON_{ADM}$ does not manage strong revocation because cascading delegations are not required. In other words, this model bases on ownership and then, owners should manage access control in regard to data their posses, either being an entire piece of data or, when co-ownership management takes place, a part of it.

VII. CONCLUSIONS

In this paper, $SoNeUCON_{ADM}$, the administrative model for $SoNeUCON_{ABC}$ usage control model, has been proposed. It supports administrative tasks concerning the identification of who is involved in administrative issues and how they are performed. $SoNeUCON_{ADM}$ has been assessed against a pair of administrative access control models (RBAC and $UCON_{ABC}$) to ensure that it successfully addresses all identified administrative tasks.

In what concerns $SoNeUCON_{ADM}$, the main future step is the management of temporal delegations. Moreover, its

implementation either in a real or in a simulated environment is expected in future work to prove the feasibility of its implementation and the study of users satisfaction.

REFERENCES

[1] L. González−Manzano, A. I. González−Tablas, J. M. de Fuentes, and A. Ribagorda, "SoNeUCON$_{ABC}$, an expressive usage control model for Web-Based Social Networks," *Computers & Security, In Press*, 2014.

[2] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models," *Computer*, vol. 29, no. 2, pp. 38 –47, 1996.

[3] R. Sandhu, V. Bhamidipati, and Q. Munawer, "The arbac97 model for role-based administration of roles," *ACM Trans. Inf. Syst. Secur.*, vol. 2, no. 1, pp. 105–135, 1999.

[4] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham, "A semantic web based framework for social network access control," in *Proceedings of the 14th ACM symposium on Access control models and technologies*, ser. SACMAT '09. ACM, 2009, pp. 177–186.

[5] M. Thompson, A. Essiari, and S. Mudumbai, "Certificate-based authorization policy in a pki environment," *ACM Trans. Inf. Syst. Secur.*, vol. 6, no. 4, pp. 566–588, 2003.

[6] A. Squicciarini, M. Shehab, and F. Paci, "Collective privacy management in social networks," in *Proceedings of the 18th international conference on World wide web*, ser. WWW '09, 2009, pp. 521–530.

[7] A. Squicciarini, M. Shehab, and J. Wede, "Privacy policies for shared content in social network sites," *The VLDB Journal*, vol. 19, no. 6, pp. 777–796, 2010.

[8] A. Ahmad, B. Whitworth, and L. Janczewski, "More choices, more control: Extending access control by meta-rights reallocation," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on*, 2012, pp. 1113 –1118.

[9] Y. Jung and J. Joshi, "Cpbac: Property-based access control model for secure cooperation in online social networks," *Computers & Security*, 2013.

[10] Y. Ren, "Access control in a cooperative editing system," in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, 2011, pp. 77 –80.

[11] M. Prilla and C. Ritterskamp, "Collaboration support by co-ownership of documents," in *Proceedings of the 2006 conference on Cooperative Systems Design: Seamless Integration of Artifacts and Conversations – Enhanced Concepts of Infrastructure for Communication*, 2006, pp. 255–269.

[12] A. Imine, A. Cherif, and M. Rusinowitch, "A flexible access control model for distributed collaborative editors," in *Proceedings of the 6th VLDB Workshop on Secure Data Management*, ser. SDM '09, 2009, pp. 89–106.

[13] M. Lorch, D. B. Adams, D. Kafura, M. S. R. Koneni, A. Rathi, and S. Shah, "The PRIMA System for Privilege Management, Authorization and Enforcement in Grid Environments," in *Proceedings of the 4th International Workshop on Grid Computing*, ser. GRID '03. IEEE Computer Society, 2003, pp. 109–.

[14] H. Wedde and M. Lischka, "Cooperative role-based administration," in *Proceedings of the eighth ACM symposium on Access control models and technologies*. ACM, 2003, pp. 21–32.

[15] R. Sandhu, R. Krishnan, J. Niu, and W. Winsborough, "Group-centric models for secure and agile information sharing," *Computer Network Security*, pp. 55–69, 2010.

[16] R. Sandhu, K. Bijon, X. Jin, and R. Krishnan, "Rt-based administrative models for community cyber security information sharing," in *CollaborateCom*, 2011, pp. 473–478.

[17] W. Edwards, "Policies and roles in collaborative applications," in *Proceedings of the 1996 ACM conference on Computer supported cooperative work*, ser. CSCW '96. ACM, 1996, pp. 11–20.

[18] K. Sikkel, "A group-based authorization model for cooperative systems," in *Proceedings of the fifth conference on European Conference on Computer-Supported Cooperative Work*, ser. ECSCW'97. Kluwer Academic Publishers, 1997, pp. 345–360.

[19] Z. Zhang, T. Huang, Q. Wu, and J. Pu, "A cscw-enabling integrated access control model and its application," *Key Engineering Materials*, vol. 460, pp. 96–105, 2011.

[20] R. Thomas, "Team-based access control (tmac): a primitive for applying role-based access controls in collaborative environments," in *Proceedings of the second ACM workshop on Role-based access control*, ser. RBAC '97. ACM, 1997, pp. 13–19.

[21] E. Cohen, R. Thomas, W. Winsborough, and D. Shands, "Models for coalition-based access control (cbac)," in *SACMAT*, 2002, pp. 97–106.

[22] V. Gligor, H. Khurana, R. Koleva, V. Bharadwaj, and J. Baras, "On the negotiation of access control policies," in *Security Protocols*. Springer, 2002, pp. 188–201.

[23] J. Jin and G.-J. Ahn, "Role-based access management for ad-hoc collaborative sharing," in *Proceedings of the eleventh ACM symposium on Access control models and technologies*, ser. SACMAT '06. ACM, 2006, pp. 200–209.

[24] H. Zhang, W. Wu, and Z. Li, "Open social based group access control framework for e-science data infrastructure," in *E-Science (e-Science), 2012 IEEE 8th International Conference on*. IEEE, 2012, pp. 1–8.

[25] B. Carminati, E. Ferrari, R. Heatherly, M. Kantarcioglu, and B. Thuraisingham, "Semantic web-based social network access control," *computers & security*, vol. 30, no. 2, pp. 108–115, 2011.

[26] P. W. Fong and I. Siahaan, "Relationship-based access control policies and their policy languages," in *Proceedings of the 16th ACM symposium on Access control models and technologies*, ser. SACMAT '11. ACM, 2011, pp. 51–60.

[27] Y. Cheng, J. Park, and R. Sandhu, "Relationship-Based Access Control for Online Social Networks: Beyond User-to-User Relationships," in *SocialCom*, 2012, pp. 646–655.

[28] B. Carminati, E. Ferrari, and A. Perego, "Rule-Based access control for social networks," in *Proceedings of the 2006 international conference on On the Move to Meaningful Internet Systems: AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET - Volume Part II*, ser. OTM'06. Springer-Verlag, 2006, pp. 1734–1744.

[29] B. Carminati and E. Ferrari, "Access control and privacy in web-based social networks," in *International Journal of Web Information Systems*, no. 4, 2008, pp. 395–415.

[30] A. Lazouski, F. Martinelli, and P. Mori, "Usage control in computer security: A survey," *Computer Science Review*, vol. 4, no. 2, pp. 81 – 99, 2010.

[31] N. Attrapadung and H. Imai, "Attribute-based encryption supporting direct/indirect revocation modes," in *Cryptography and Coding*. Springer, 2009, pp. 278–300.

[32] J. Park, "Usage Control: A Unified Framework for Next Generation Access Control," Ph.D. dissertation, George Mason University, 2003.

[33] F. Salim, J. Reid, and E. Dawson, "An administrative model for UCON$_{ABC}$," in *Proceedings of the Eighth Australasian Conference on Information Security*, ser. AISC '10, 2010, pp. 32–38.

[34] R. Sandhu and J. Park, "Usage control: A vision for next generation access control," *Computer Network Security*, pp. 17–31, 2003.

[35] J. Crampton and H. Khambhammettu, "Delegation in role-based access control," in *Computer Security–ESORICS 2006*. Springer, 2006, pp. 174–191.

[36] Z. Zhang, L. Yang, Q. Pei, and J. Ma, "Research on usage control model with delegation characteristics based on om-am methodology," in *Network and Parallel Computing Workshops, 2007. NPC Workshops. IFIP International Conference on*. IEEE, 2007, pp. 238–243.

La ventana de AREM. Una herramienta estratégica y táctica para visualizar la incertidumbre

Jeimy J. Cano M.
Facultad de Derecho
Universidad de los Andes
Bogotá, Colombia
Email: jcano@uniandes.edu.co

***Resumen*—Los fundamentos de la administración de riesgos tradicional poco a poco se han venido debilitando frente a la dinámica, incertidumbre y ambigüedad de los entornos de negocio. Lo que antes era medianamente viable anticipar, ahora es prácticamente imposible establecer un análisis certero que permita ofrecer orientación y claridad sobre qué hacer frente a una situación particular. En este sentido, se introduce la ventana de AREM, cuyo objetivo es ampliar la capacidad de conocimiento del entorno y facilitar una toma de decisiones informada sobre las oportunidades y retos empresariales, buscando ajustar la práctica actual de la administración de riesgos en un mundo dominado por las redes sociales, la computación en la nube, los dispositivos móviles y la información instantánea.**

***Palabras clave*—Riesgos, amenazas, decisiones informadas, incertidumbre, entorno dinámico, Ventana de Johari**

I. Introducción

Muchos investigadores y ciudadanos pasan un tiempo importante pensando sobre el futuro, sobre las condiciones y riesgos que están por venir, sobre los nuevos retos que se van a plantear y la forma como deberemos enfrentarlos para alcanzar nuevas posiciones estratégicas en los mercados actuales y futuros.

Conocer el futuro es un problema complejo, pues exige saber que está pasando en la actualidad con certeza y ver entre líneas los aspectos emergentes, que pueden ser hoy sólo movimientos aparentemente pasajeros, pero que no dimensionamos sus alcances o factores que los puedan potenciar. Mucho del conocimiento que podamos tener del entorno no bastará para establecer un patrón de conocimiento emergente que pueda revelar el misterio lo que puede pasar en el mediano y largo plazo.

En este sentido, las organizaciones deben establecer estrategias que permitan cada vez más avanzar en el conocimiento de su entorno, buscando un recorrido perimetral completo que disminuya los puntos ciegos y aumentar su capacidad de anticipación, toda vez que es de esta forma como puede permanecer vigente y competitiva, aún frente a situaciones inesperadas.

Como quiera que el futuro, es una reflexión basada en escenarios aún por conocer o mejor, situaciones que ya pasaron y no las vimos, se hace necesario mantener en el radar una forma de afinar la revisión del medio ambiente empresarial y las situaciones propias de cada organización, con el fin de tener a la vista motivaciones y sugerencias que nos permitan ubicar los riesgos y amenazas emergentes más relevantes y poder actuar en consecuencia.

Habida cuenta de lo anterior, utilizando los conocimientos y estrategias fundadas en las relaciones humanas, asociadas con la forma de ofrecer retroalimentación, como lo es la Ventana de Johari, instrumento ampliamente conocido en el ámbito social, buscaremos repensar los conceptos de amenazas y riesgos emergentes en el contexto empresarial, para ofrecer una nueva lectura de la Ventana de Johari, ahora como la Ventana de AREM (Amenazas y Riesgos Emergentes).

Tabla I
La Ventana de Johari (Tomado de: [2]

	Conocido por la persona	**Desconocido por la persona**
Conocido por los otros	ÁREA LIBRE	ÁREA CIEGA
Desconocido por los otros	ÁREA OCULTA	ÁREA DESCONOCIDA

II. Entendiendo la ventana de Johari

"Todos los seres humanos tenemos zonas abiertas, zonas ocultas, zonas ciegas y zonas desconocidas. Mientras más amplia sea la zona abierta, mayor es la apertura y transparencia que generamos cuando interactuamos con los demás. Estas diferentes zonas fueron detalladas por los psicólogos Joseph Luft y Harry Ingham, que denominaron la Ventana de Johari, como una estrategia para mejorar la comunicación a través de la cual una persona da o recibe informaciones sobre sí misma o sobre otras personas.

La idea de usar este instrumento es alertar y tratar las áreas ocultas y ciegas que son las que predominan en la relación interpersonal. El área oculta es aquella donde hay elementos desconocidos por los demás y conocidos por la persona, mientras el área ciega, es aquello que la persona no conoce, pero es conocido por los demás." [1].

La ventana de Johari, detalla FRITZEN [2] (pág.11), "trata de explicar cómo deben procurar tolerarse mutuamente las diferencias en las distintas áreas de nuestra personalidad, con el fin de mejorar las relaciones interpersonales, a través del conocimiento de uno mismo y de los demás. (...)". Este

instrumento abre la posibilidad para explorar campos desconocidos de nuestro propio entorno que lleven a una mayor expansión del área libre, donde podemos actuar y movilizarnos para hacer de nuestras relaciones con otros, fuentes de ventaja emocional y empresarial, donde podemos construir propuestas conjuntas y relaciones gana-gana que superen la lucha de egos y se privilegien la suma de talentos y voluntades.

Así las cosas, la ventana de Johari es una forma para aumentar nuestra transparencia en las comunicaciones, motivando la posibilidad de aprender con los otros y recomponer el campo de las zonas grises de las relaciones humanas, que en lugar de ser espacios para la confrontación y la contienda, se conviertan en oportunidades concretas y confiables para que surja la confianza, esa condición fundamental para hacer que las cosas pasen.

III. Las comunicaciones entre la empresa y su entorno: riesgos, amenazas y oportunidades

Si bien la Ventaja de Johari es un instrumento propio de las ciencias sociales, pensar en su aplicación a nivel empresarial resulta algo extraño y hasta incómodo, toda vez que pensar en comprender la misma dinámica entre la empresa y el entorno, como una persona y un grupo, puede ser algo inicialmente inesperado, peligroso y retador.

En el mundo empresarial, los análisis de las oportunidades de negocio siguen modelos establecidos y condiciones particulares que generan informes tipo que responden más o menos a lo que los ejecutivos quieren escuchar o entender. Rara vez, los informes de los analistas de entorno sugieren elementos que no estén dentro de los parámetros propios de sus pronósticos, como quiera que hacerlo, implica compromiso de su credibilidad y debilitamiento de su imagen frente a las decisiones que los ejecutivos van a tomar siguiendo sus reflexiones.

Así las cosas, comprender el entorno empresarial y revelar los patrones de cambio que se avecinan, no es sólo un ejercicio de correlación de eventos y datos acumulados durante mucho tiempo, sino de lanzarse a ver aquellos "espacios en blanco" que sugieren los números, situaciones excepcionales, condiciones límites, manifestaciones impropias y frecuentemente peligrosas que permite a la empresa, cambiar las cosas y quebrar los lentes de su inercia sectorial, para superar sus temores frente a la competencia y crear una nueva realidad donde, como afirma REHN [3, pág. 34], pocos se puedan aventurar y donde sus técnicas de navegación mental ya no sean aplicables.

En este contexto, mientras más sea el conocimiento de la empresa sobre sus capacidades y habilidades para comprometer y desestabilizar su entorno, mayor será su fortaleza para transformar su ambiente y revelar las nuevas condiciones, que cambien la manera de hacer las cosas. De igual forma, cuando somos capaces de profundizar en las variaciones del entorno y prever nuevos vectores de riesgos, estamos avanzando en una reflexión que le permite a la empresa anticiparse y caminar sobre las aguas inciertas, con la vista en el nuevo reto, que sabe que enfrentará y superará, pues ha visto sus manifestaciones invisibles y advierte sus posibles impactos.

Por tanto, ante este escenario retador y de permanente incertidumbre, se requiere, como afirma ÁLVAREZ DE MON [4] (pág.18) "un carácter decidido y resuelto, curtido en multitud de ocasiones, enfrentado a dilemas críticos (...)" que permita a los ejecutivos de las empresas de hoy, reconocer en la inestabilidad de los mercados, esa forma natural de "sentirse perdido" como prerrequisito natural para encontrarse consigo mismo y descubrir su auténtica realidad, esa que está más allá de los ejercicios de planeación estratégica, que se esconde en la manera como la organización se relaciona con su propia realidad, sin entrar como afirma ÁLVAREZ DE MON [4] (pág.19) en un superávit de futuro que genere angustia e intranquilidad insoportable.

IV. La ventana de AREM. Una lectura de la incertidumbre en las relaciones de la empresa y su entorno

Modelar las relaciones emergentes del entorno de negocio es una actividad que reta cualquier modelo de pronóstico disponible a la fecha, pues tratar de representar la realidad futura con escenarios, siempre tendrá la limitación de la variable que no se puede especificar. Sin embargo, tratar de establecer un marco general de amenazas y riesgos emergentes (AREM), puede ser una lectura que permita a las organizaciones clasificar mejor sus acciones y tratar de equivocarse de manera diferente, motivando un "pensamiento peligroso" que evoluciona desde los tradicionales riesgos conocidos hasta las propuestas con momentos y condiciones inesperadas y desconocidas que, al contar con suficientes enemigos o detractores [3, pág.34], valga la pena el ejercicio de oposición y crítica.

Así como, la Ventana de Johari, es un instrumento que nos permite establecer estrategias para aumentar la zona libre, esa área de conocimiento mutuo y convergencia de informaciones entre el yo y los demás, se propone hacer una lectura equivalente entre la empresa y su entorno, con el fin de crear un instrumento que mantenga en la mira de los ejecutivos de la empresa, los diferentes tipos de riesgos y amenazas que debe atender si quiere mantenerse vigente en su entorno.

En este sentido, se plantea la Ventana de AREM, como una vista estratégica y táctica de actuación de los ejecutivos de la empresa, para comprender los aspectos conocidos y desconocidos de sus capacidades empresariales, en el contexto de aquellos riesgos y amenazas propias de su entorno, así como de los vectores de inestabilidades emergentes que deben identificar dentro y fuera de su realidad corporativa.

Siguiendo esta propuesta, se presenta a continuación la Ventana de AREM que establece las diferentes zonas de alcance de las amenazas y riesgos conocidos, focalizados, latentes y emergentes como una forma de motivar el análisis de los ejecutivos y provocar las ideas que "en tiempos de guerra" permitan encontrar maneras de sobrevivir.

Las amenazas y riesgos conocidos son las situaciones tradicionales que se presentan en la organización. Las típicas sesiones de riesgos que buscan comprender que acciones se

Tabla II
LA VENTANA DE AREM

	Conocido por la empresa	**Desconocido por la empresa**
Conocido por el entorno	AMENAZAS Y RIESGOS CONOCIDOS	AMENAZAS Y RIESGOS LATENTES
Desconocido por el entorno	AMENAZAS Y RIESGOS FOCALIZADOS	AMENAZAS Y RIESGOS EMERGENTES

adelantan frente a temas como fuga de información, errores y/o omisiones, multas y/o sanciones, pérdidas de imagen, acceso no autorizado, pérdida de liquidez, inestabilidad de mercados, entre otros que mantienen la operación de la empresa frente a condiciones que se sabe conocidas y que exigen acciones permanentes y concretas (planes de tratamiento de riesgo) para mantenerlas bajo control y conocer qué hacer cuando alguno de ellos se materializa.

Las amenazas y riesgos latentes, son aquellas circunstancias que se advierten en el entorno y que son generalmente desconocidas en el contexto de negocio de las empresas. Dichas situaciones se manifiestan de manera frecuente y son detectadas por los analistas empresariales, sugiriendo patrones y condiciones particulares a través de la cual se pueden presentar. Su comportamiento es asimétrico y genera ruido e incertidumbre en el entorno, condiciones propias de una sorpresa predecible. Un ejemplo de esta realidad la podemos asociar con los patrones de comportamiento que se identificaron con la deuda inmobiliaria en los Estados Unidos de América que generó una crisis interna con afectación internacional, la cual se advertía en los análisis de los estudiosos de los mercados.

Las amenazas y riesgos focalizados, son situaciones propias de una industria o sector de negocio, que afectan la competitividad y posicionamiento de una empresa en un nicho específico. Son escenarios por lo general desconocidos en el entorno, pero conocidos por la empresa, dado que reconoce y sabe el comportamiento de su sector de negocio. Este tipo de amenazas y riesgos focalizados, deben instar a los ejecutivos de la empresa a estudiar de manera permanente los avances y novedades de su industria, para identificar nuevas formas de hacerlas cosas y crear los escenarios donde su empresa pueda movilizarse sobre las inestabilidades del sector y alcanzar una posición privilegiada.

Finalmente y no menos importantes, las amenazas y riesgos emergentes, esos momentos, ideas, propuestas de contextos que retan las capacidades de la empresa y privilegian más las posibilidades que las probabilidades. En este sentido, como afirman WESTERMAN y HUNTER [5] (pág.22) “las empresas responden de manera más eficiente a los riesgos que entienden, por muy impredecibles que sean, que a los que no entienden”. Así las cosas, situaciones como las amenazas persistentes avanzadas y los ataques ciberterroristas, las podríamos tipificar como condiciones básicas para pensar e investigar en escenarios aún no conocidos, que suponen per se cambios o situaciones disruptivas.

V. REVISANDO LA VENTANA DE AREM. UNA LECTURA DESDE LA INSEGURIDAD DE LA INFORMACIÓN

Los analistas de Mckinsey [6] establecen diez tendencias emergentes que transformarán la vida personal y empresarial en la próxima década. Dicha lista está conformada por:

* Las tecnologías sociales, aquellas que potencian la interacción de los individuos y organizaciones en la red;

* El uso de grandes volúmenes de datos y su respectiva analítica, con el fin de advertir patrones emergentes y relaciones desconocidas reveladas a través de los datos;

* El internet de las cosas, como una nueva realidad extendida ahora con dispositivos cotidianos conectados a la red;

* La propuesta de “todo como un servicio” como una forma de estandarizar todo aquello que podemos hacer con recursos compartidos y visibles en la red;

* La automatización del conocimiento, que potencia a los trabajadores del conocimiento, que aumente la productividad empresarial más allá de los patrones identificados en el ejercicio de la analítica de los datos;

* La potenciación del ciudadano digital, como factor diferenciador de un nativo digital informado y dispuesto a transformar la manera de hacer las cosas;

* El quiebre de las fronteras entre lo físico y lo virtual, como una renovación del espacio vital del ser humano, ahora con experiencias diferentes y aumentadas de su realidad;

* La personalización y simplificación de los modelos de negocio basados en la red, como una vista particularizada de los intereses de los ciudadanos digitales, clientes de nuevos bienes y servicios;

* La creación de mercados electrónicos dirigidos por las tendencias de los nativos digitales que permiten la evolución de los sistemas pagos y logística en el entorno digital y

* Los impactos de los avances tecnológicos en el gobierno, la educación y la salud, como forma de favorecer el producto interno bruto de las naciones que desean ser parte de la realidad de un mundo en línea y con límites desconocidos.

Revisando cada una de estas tendencias encontramos un común denominador en cada una de ellas. Un elemento que permite cohesionar cada una de ellas y diferenciarlas al mismo tiempo. Una fuente de oportunidades y riesgos propios que busca crear una sensación de exclusividad para todos aquellos que participan de un entorno digital, altamente interconectado, con información instantánea y ahora en la nube. La información se convierte entonces, en palabras de un académico colombiano, en el nuevo “petróleo del siglo XXI”, una materia prima que está en capacidad de generar valor para aquellos que logran transformarla en activos valiosos, por los cuales otros están dispuestos a pagar.

Habida cuenta de lo anterior, es a través de la información como las empresas logran superar sus miedos o potenciar sus temores. Es la forma como cada una de ellas es capaz de tratarla, lo que hace que algunas se ubiquen en lugares privilegiados y otras no. Por tanto, su inadecuado tratamiento puede llevar a situaciones delicadas que deterioran su capacidad de anticipar las amenazas y riesgos que afecten su operación o lo que es peor, comprometan su futuro.

En razón con lo anterior, se hace necesario pensar de manera diferente frente a los análisis de riesgos relacionados con la seguridad de la información y motivar un ejercicio desde la Ventana de AREM que "luche contra las tendencias profundamente arraigadas de nuestro pensamiento" [3] (pág.64) para abrirle las puertas a las posibilidades, considerando, como afirma TRUMP y KIYOSAKI [7] (pág.206) que tanto la empresa como su entorno son "tu propia escuela de negocios y programa de desarrollo personal".

La ventana de AREM leída en clave de la inseguridad de la información, es una herramienta de pensamiento lateral [8], que deben ser animada por el cuestionamiento de las suposiciones, hacer las preguntas correctas, intentar nuevos marcos o puntos de vista (aún suenen descabellados) y motivar un análisis lógico y disciplinado del razonamiento que interrogue la solución cómoda y conocida, para entender aquello que está en el campo de las posibilidades y no en el de las probabilidades.

VI. Incorporando la ventana de AREM

La ventana de AREM, surge como una forma diferente de motivar una reflexión sobre el escenario de análisis que se seleccione, con el fin de movilizar a las empresas fuera de su zona de confort frente a los riesgos, para sumergirla más allá de las condiciones actuales conocidas de su entorno y motivar una nueva vista de la empresa, tanto en su sector de negocio, como más allá de aquello que ella conoce y su entorno le manifiesta.

Como quiera que para adelantar este ejercicio renovado de riesgos, se hace necesario contar con personas especialistas en el tema, así como de personas fuera del dominio de estudio, de los ejecutivos de negocio y de personal externo a la empresa (si es posible), es importante documentar bien la sesión de trabajo, trayendo a la mesa de trabajo, aquellas situaciones que actualmente se advierten fruto del ejercicio tradicional de riesgos, que es donde iniciará la revisión de la problemática, para ir avanzando a los siguientes cuadrantes de la ventana.

Es importante anotar, que los profesionales que participen del ejercicio de la ventana de AREM, cuyo objetivo es ampliar la capacidad de conocimiento del entorno y facilitar una toma de decisiones informada sobre las oportunidades y retos del tema analizado, deben tener la seguridad psicológica y real que sus comentarios serán tomados como expresiones de cómo "conectar los puntos" para ver más allá de los hechos y eventos, y no como forma de atacar una forma de pensar particular, que genere contradictores que impidan que se movilicen las ideas y propuestas que deben salir de la sesión.

La ventana de AREM debe motivar a los participantes a "pensar fuera de la caja" y fortalecer las lecturas conexas de los múltiples riesgos identificados, para comenzar a mover las reflexiones desde el cuadrante de lo conocido, hacia aquello que es latente, es decir, aquello que no es evidente en el momento, pero que existen condiciones y señales en el ambiente que establecen patrones de actividad, que advierten de una situación que aparece como irrelevante, pero que existen suficientes elementos para tenerla en consideración en los análisis.

Aquellos participantes que han identificado estas nuevas aproximaciones del entorno, deben comentar y documentarlas de tal forma que describan con la mayor claridad su vista y establecer, un proceso de revisión y validación con un equipo de trabajo especializado para establecer allí sus análisis detallados y luego traerlos a la mesa, una vez se hayan estudiado.

De otra parte, los especialistas de negocio y analistas de mercados, hacen lo propio frente a los riesgos focalizados, para lo cual reconocen la nuevas tendencias de la industria relacionadas con la temática revisada, así como elementos a saber regulaciones, nuevas prácticas, estándares, movimientos políticos, sociales o empresariales, que son percibidos por la competencia en su sector o que se manifiestan como temáticas relevantes dentro de los círculos de influencia de la industria.

La consulta de los principales proveedores de la industria y analistas externos de su sector, así como estudios sectoriales son elementos relevantes para establecer aquellos elementos que son emergentes en su área de negocio, para mantenerse competitivos y con ventaja sobresaliente, siempre y cuando pueda anticiparse a las mismas o lo que es mejor, que sean sus acciones y actividades las que crean las nuevas condiciones de su entorno de negocio.

Luego de revisados y analizados los resultados en cada uno de los cuadrantes, queda expuesta la síntesis de los riesgos identificados y sus planes de tratamiento para avanzar y anticiparse a los impactos que éstos puedan tener en los planes de la empresa, en el mediano y largo plazo. Sin embargo, aún no está completo el ejercicio faltando el momento para cuestionar nuestro entendimiento actual de los riesgos y el marco conceptual en el cual han sido concebidos para evidenciar las discontinuidades y saltos inesperados de las tendencias actuales, para lo cual se hace necesario abrir el espacio a las posibilidades y dejar de pensar en las probabilidades, ampliamente conocidas para los ejercicios tradicionales de riesgos.

En el sector de amenazas y riesgos emergentes, se requiere contar con habilidades particulares de aquellos que son capaces de: [9]

1. Ver los cambios que vienen
2. Comprender las implicaciones de dichos cambios
3. Anticipar la trayectoria de los cambios

Es decir profesionales que están dedicados a vigilar y correlacionar eventos inesperados, variables asimétricas del entorno y comportamientos socio-económicos inadvertidos, tecnologías disruptivas, así como cambios en las preferencias de los diferentes grupos de interés, que permitan animar nuevas reflexiones sobre las implicaciones que se pueden generar para la organización y sus planes de mediano y largo plazo.

Las ideas que se presenten en este sector de la ventana serán el trasfondo del conocimiento de la realidad empresarial, un lugar donde buscar nuevas forma de crear valor en la

empresas, es decir, para diseñar nuevos activos no documentados, así como impactos aún no dimensionados, que para algunos podría denominarse un posible "cisne negro" que pueda tener implicaciones bien positivas o devastadoras para la organización.

Terminado el recorrido por cada uno de los segmentos de la ventana de AREM, se establecen los riesgos y amenazas emergentes más relevantes para los asistentes del ejercicio, los cuales serán documentados por cada uno de los grupos de trabajo establecidos para cada cuadrante, quienes ahora deberán desarrollar las estrategias de tratamiento de los mismos y entregar una vista consolidada de cada segmento de la ventana, para que la empresa cuente con acciones concretas y análisis claves para documentar sus decisiones.

VII. Conclusión

Anota ÁLVAREZ DE MON [4] (pág.18) "las cumbres del espíritu humano han sido conquistadas a base de ensayo y error. (...) un itinerario que dista mucho de ser sencillo. (...)", palabras que recogen muy bien las reflexiones que se han planteado alrededor del ejercicio de tratar de aventurarnos en las aguas de la incertidumbre. No podemos negar que nuestro conocimiento finito e incipiente de la realidad nos margina muchas veces de nuevas oportunidades para crear activos claves que representen valor y beneficio para un tercero.

Sin perjuicio de lo anterior, existen iniciativas que buscan enfrentarse al reto de entender las condiciones del entorno, aun sabiendo que para sobrevivir en un sector de negocios, se requiere como afirma TRUMP y KIYOSAKI [7] (pág.228-229) "aprender de distintos temas y hacerlo con rapidez", so pena de ser arrasados por la constante variación de los mercados y de los sectores de negocio. En este sentido, la ventana de AREM, es una propuesta novedosa que busca motivar a los ejecutivos para pensar sobre aquello que conocen y desconocen, una apuesta para detallar y profundizar la misión empresarial que los moviliza para crear las condiciones de operación que les permita "caminar sobre las aguas" de la inestabilidad y no morir en el intento.

De igual forma, la ventana de AREM es una forma de nunca subestimar las condiciones asimétricas de la inseguridad de la información, sino más bien, de pensar en los detalles que implican su entendimiento, la mente de los atacantes, las relaciones propias entre la tecnología, los procesos y las personas, las vulnerabilidades latentes, que no marginan el conocimiento de los actores de la organización, sino que potencian sus reflexiones para crear un vitrina de aprendizaje y desaprendizaje que hablan de una empresa resistente a la fallas no por excepción, sino por convicción.

Si en el ejercicio de proteger los activos de información de la empresa, los ejecutivos y el responsable de seguridad se concentran en los mecanismos de prevención y control, no estarán actuando en el escenario real de su entorno y el ejercicio de la ventana de AREM será marginado a listas de chequeo que no desarrollarán su capacidad proactiva. Mientras, si estos actores se enfocan en el conocimiento abierto y concreto de su escenario de amenazas y riesgos emergentes, con una vista diligente y creativa, estaremos creando un activo intangible de protección que está más allá de una estrategia de seguridad de la información, una capacidad de actuación que moviliza decisiones sencillas y ejecutables.

Recuerde que en la era de la información y el conocimiento, afirman TRUMP y KIYOSAKI [7] (pág.253) "no es la velocidad de las líneas de ensamblaje la clave del éxito, sino la velocidad y la alta calidad con que el pensamiento humano trabaja para alcanzar un objetivo común", por tanto que la ventana de AREM, se transforme poco a poco en esa forma de ver en el margen de las hojas y la excusa para detectar los detalles que hacen la diferencia, cuando nuestro pensamiento insista en retornar a su zona de confort.

Agradecimientos

El autor agradece el apoyo y los acertados comentarios del doctor Jorge Ramio Aguirre, la maestra Gabriela Saucedo Meza, el maestro Andrés Almanza Junco y los ingenieros Alberto León Lozano, Daliris Maldonado Gómez y Mauricio Luna Salguero, que permitieron refinar las ideas de este artículo.

Referencias

[1] CANO, J. (2013) Comunicación con DIOS. Frase de la Semana. Disponible en: http://frasedelaseman.blogspot.com/2013/05/comunicacion-con-dios.html (Consultado: 3-06-2013)

[2] FRITZEN, S. (1987) *La ventana de Johari. Ejercicios de dinámica de grupo, de relaciones humanas y de sensibilización*. Sal Terrae.

[3] REHN, A. (2012) *Ideas peligrosas. Cuando el pensamiento provocador se convierte en el activo más valioso*. Pearson.

[4] ÁLVAREZ DE MON, S. (2009) Incertidumbre, hábitat natural del directivo. *IESEInsight*. No.1. Segundo trimestre.

[5] WESTERMAN, G. y HUNTER, R. (2009) Hable un idioma común sobre el riesgo en TI. *IESEInsight*. No.1. Segundo trimestre.

[6] BUGHIN, J. , CHUI, M. y MANYIKA, J. (2013) Ten IT-enabled business trends for the decade ahead. Mckinsey Quarterly. May. Disponible en: http://www.mckinsey.com/insights/high_tech_telecoms_internet/ten_it-enabled_business_trends_for_the_decade_ahead (Consultado: 3-06-2013)

[7] TRUMP, D. y KIYOSAKI, R. (2013) *El toque de Midas. Por qué algunos empresarios se hacen ricos, pero la mayoría no*. Ed. Aguilar.

[8] DE BONO, E. (2006) *El Pensamiento Lateral*. Editorial Paidós Ibérica S.A.

[9] FUNSTON, F. y WAGNER, S. (2010) *Surviving and Thriving in uncertainty. Creating the risk intelligent Enterprise*. John Wiley and Sons.

Seguridad en smart cities e infraestructuras críticas

Victor Garcia-Font
Internet Interdisciplinary Institute
Universitat Oberta de Catalunya
Email: vgarciafo@uoc.edu

Carles Garrigues
Estudis d'Informàtica Multimèdia i Telecomunicació
Universitat Oberta de Catalunya
Email: cgarrigueso@uoc.edu

Helena Rifà-Pous
Estudis d'Informàtica Multimèdia i Telecomunicació
Universitat Oberta de Catalunya
Email: hrifa@uoc.edu

***Resumen*—Numerosas ciudades están desarrollando plataformas smart city con el fin de lograr una mejor coordinación, eficacia, reducción de costes y en general una gestión más eficiente de la ciudad, a través de la integración de infraestructuras y servicios. Entre los subsistemas que se integran en una smart city hay un grupo de especial interés formado por las infraestructuras críticas. Con la integración de estas se busca dar un mejor servicio, pero al aumentar la complejidad y la dependencia de unas infraestructuras con otras y con las TIC, crece el riesgo de que una vulnerabilidad o fallo en una infraestructura pueda extenderse y ocasionar fallos en otra, y así sucesivamente provocando un fallo en cascada.**

En este artículo describimos un diagrama común de muchos proyectos para smart cities y analizamos los problemas de seguridad y privacidad que aparecen al interconectar las infraestructuras y al tender hacia una filosofía de datos abiertos.

***Palabras clave*—ciberseguridad *(cybersecurity)*, ciudad inteligente *(smart city)*, datos abiertos *(open data)*, infraestructura crítica *(critical infrastructure)*, privacidad *(privacy)*, Tecnologías de la Información y la Comunicación *(Information and Communication Technologies)*.**

I. Introducción

En los últimos años, las ciudades han añadido a los retos clásicos nuevos desafíos propios de la sociedad contemporánea: absorción del aumento de la población, reducción del consumo energético y de emisiones de CO2, mayor sostenibilidad, crecimiento económico, etc. Para hacer frente a estos puntos se está produciendo un progresivo aumento de la inversión en capital humano y tecnológico. Usando las TIC como base, llamamos smart city a las ciudades que buscan atajar estos desafíos desarrollando sistemas para la mejora en áreas como la gobernanza, la energía, el medio ambiente, la movilidad o la economía entre otros. El desarrollo de estos sistemas conlleva inherentemente nuevos modelos de operación y modifica características básicas de las ciudades. La implantación de líneas de telecomunicaciones interconecta más a los ciudadanos con las instituciones, a las empresas con sus proveedores, a los gestores de infraestructuras con las infraestructuras que gestionan, etc. Además, esta interconexión también se produce con elementos insertados en el entorno urbano, como por ejemplo cámaras de video vigilancia, sensores, teléfonos móviles, o dispositivos GPS, que generan una gran cantidad de información que pasa a estar disponible no sólo localmente sino a una escala mayor para el conjunto de los entes que conforman la ciudad. Todo esto provoca que se demanden nuevos servicios, se abran nuevas líneas de negocio, se creen nuevos empleos, se puedan automatizar operativas urbanas y mejorar la eficiencia en la gestión de infraestructuras, se haga la ciudad más competitiva y se potencie más transparencia en la gestión pública.

Todos estos cambios que aporta la smart city también afectan a las que llamamos infraestructuras críticas. Éstas son las instalaciones clave que proporcionan los servicios que afectan al bienestar de las personas, sea suministrando directamente estos servicios esenciales, o dando servicio a otra infraestructura crítica para que ésta pueda operar correctamente. Más concretamente, las infraestructuras críticas más destacadas están relacionadas con la energía eléctrica, la producción y distribución de combustibles, las telecomunicaciones, el transporte, la distribución de agua, la agricultura, la banca y las finanzas, los servicios de emergencia y gobernanza, la educación y la sanidad entre otros [1].

Más integración e interconexión trae consigo una mayor complejidad y un mayor riesgo de vulnerabilidades. En este artículo vemos como la implantación de las smart cities abre brechas en la seguridad de la información y en la privacidad de los usuarios. Para empezar, hacemos una descripción a alto nivel de los sistemas de información que se están diseñando para implementar una arquitectura smart city. Posteriormente, vemos como las infraestructuras críticas usan las TIC para interconectarse, integrarse en la smart city y como estas infraestructuras dependen unas de las otras para poder operar. A continuación, revisamos cómo la seguridad informática y la privacidad pueden afectar a la construcción de una smart city. Finalmente señalamos algunos de los problemas que continúan abiertos en estos ámbitos.

II. Smart cities y infraestructuras críticas

II-A. Sistemas de información de una smart city

Los SI de una smart city son el conjunto de software, hardware y estándares que hacen posible la gestión eficiente e inteligente de la ciudad a través de las TIC. En la actualidad, hay diversas compañías y ciudades que han propuesto esquemas de arquitectura smart city. A continuación, mencionamos algunos de los productos que se están construyendo:

The PlanIT Urban Operating System[2] es una implementación de un sistema operativo para entornos urbanos que provee de tecnología en tiempo real de sensores, control, análisis espacial, integración de datos, seguridad, soporte y provisionamiento de contexto de ubicuidad para aplicaciones

del internet de las cosas (IoT). Se caracteriza por tener 4 capas: una red de sensores, una capa de control de latencia mínima para el control de los sensores, una capa de supervisión a un nivel más alto y una capa de aplicaciones. Este esquema sigue el paradigma SOA facilitando la creación de aplicaciones que usen sus servicios y la integración de módulos de otros fabricantes.

La ciudad de Oulu en Finlandia [3] ha implementado un middleware para ser usado como campo de pruebas real para mejorar y facilitar la comunicación entre los ciudadanos y el gobierno. Este middleware es una capa encima de la red LAN/bluetooth/wireless de la ciudad para facilitar el acceso a esta red y a los datos generados por sensores distribuidos por el área urbana.

En Corea del Sur están implementando el proyecto *Ubiquitous city* (u-city)[4] en el que ofrecen servicios interconectados distribuidos por áreas de interés: automatización de edificios (u-life), servicios relacionados con los negocios (u-business), gobernanza (u-government), etc. Uno de los puntos principales del proyecto es centrarse en el usuario, ofreciéndole los servicios en cualquier parte pero sin resultar intrusivo.

Chen[5] propone una arquitectura en 4 capas para la integración de la Internet of Things (IoT) en las smart cities. La capa más baja es una red de sensores autónomos que responden a estímulos del mundo real y que interactúan entre ellos. La siguiente capa es un middleware orientado a servicio que sirve como punto de unión entre los sensores y el sistema. La siguiente capa intermedia es una capa de procesamiento de datos. Hay que destacar que se proponen instrumentos para que los diferentes elementos colaboren entre sí para un procesamiento más eficaz. Por ejemplo, un smartphone de poca potencia enviaría parte de un proceso a computar al cloud. Finalmente, se propone una capa de aplicaciones y servicios.

La ciudad de Barcelona está desarrollando un esquema con la integración de varios proyectos [6], [7], [8], [9] cofinanciados por la ciudad y por otras instituciones como la Unión Europea. La capa central de middleware la constituye el *CityOS*, una agregación de módulos para procesamiento, análisis, gestión de datos históricos, BI, etc.

Entre estos módulos dentro del *CityOS* de Barcelona se encuentra el *City Service Development Kit* (*CitySDK*)[6]. Este proyecto tiene el objetivo de ayudar a las ciudades a abrir sus datos dando un conjunto de herramientas open source para facilitar a los desarrolladores la creación de servicios digitales para la ciudad. Estas herramientas son básicamente servicios digitales abiertos e interoperables, procesos, guías y estándares de usabilidad. *CitySDK* no es solamente un módulo integrado en la smart city de Barcelona, sinó que también se ha integrado en la arquitectura de otras ciudades y busca ser una pieza para que cualquier ciudad europea pueda ofrecer sus datos a desarrolladores para la creación de aplicaciones y así contribuir a la creación de una infraestructura sostenible de apps. Como colofón del proyecto, se desarrollan tres aplicaciones en los ámbitos de la participación ciudadana, el turismo y la movilidad integrados en las smart cities de Helsinki, Lisboa y Amsterdam.

En general, todas las propuestas existentes en smart cities tienen una arquitectura orientada a servicio (SOA) con una pieza central que actúa de middleware y que en muchas de las propuestas se equipara a un sistema operativo con un ámbito de ciudad. En la figura 1 se puede ver un diagrama general de bloques para una solución de este tipo. Básicamente se trata de arquitecturas en tres capas: capa de aplicaciones, capa de proceso y capa de contacto con el medio.

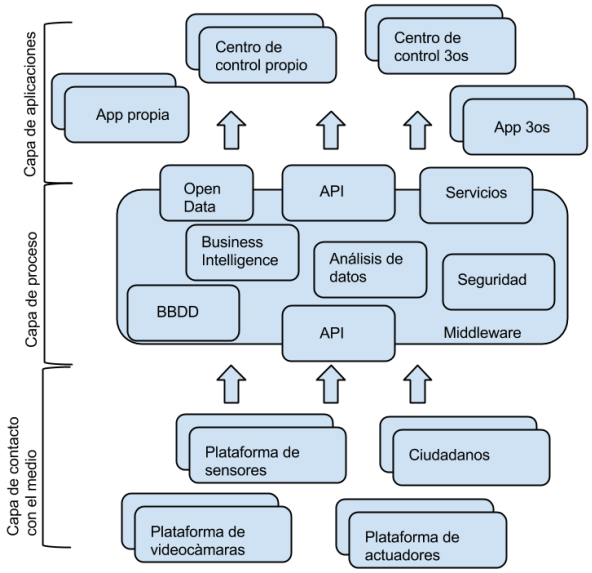


Figura 1. Diagrama de bloque general de una smart city

En la capa de aplicaciones situamos todos los elementos que usan los servicios y la información publicada por la smart city. Por ejemplo, centros de control o aplicaciones.

La capa central se trata de un middleware compuesto por muchos módulos de diversos tipos, diferentes funcionalidades e implementados por entidades diferentes que se comunican entre ellos con el uso de estándares, como por ejemplo el API REST. Las soluciones de smart city, tanto las basadas en open source como las propietarias hacen hincapié en el uso de estándares y la interoperabilidad ya que la finalidad de los sistemas de este tipo es la concentración de subsistemas y la generación de nuevos datos para ser usados en aplicaciones de diversa índole. Generalmente, el middleware aparte de ser el nexo de unión entre sensores, actuadores y aplicaciones, también es una pieza para el procesamiento, el almacenaje, la gestión y el análisis de datos. Gracias a la interoperabilidad de estos sistemas se busca poder encajar cualquier tipo de producto que dé estos servicios, como sistemas de ERP, de Business Intelligence, sistemas gestores de bases de datos o minerías de Big Data.

La capa de contacto con el medio corresponde a los elementos que alimentan con información al sistema o que el sistema usa para interactuar con los componentes de la ciudad, como por ejemplo los sensores, los actuadores o los propios ciudadanos. Dentro de esta capa, destacamos los módulos de

plataforma de sensores. En este sentido, el proyecto de open source *Sentilo*[8], implementado y desplegado en la ciudad de Barcelona, ofrece una API REST en la que se suscriben los sensores y los actuadores. Así, los sensores envían la información recogida a la plataforma de sensores para que sea procesada por agentes del sistema y reenviada hacia una capa superior o para que sirva de información de reentrada. Las funciones de suscripción y notificación crean un medio para la conexión de sensores y actuadores escalable.

II-B. Las infraestructuras críticas de una smart city

De entre todas las infraestructuras críticas, en este artículo nos centramos en las que tienen un impacto mayor en las ciudades, y por lo tanto, las que entran en el ámbito y son candidatas a ser integradas en un esquema de smart city. Más concretamente, dentro de una ciudad las infraestructuras críticas más relevantes son las energéticas, las de telecomunicaciones, la distribución de agua, la gobernanza, los servicios de emergencia y seguridad pública, el transporte y la sanidad[10].

Estas infraestructuras críticas han ido evolucionando su manera de operar y sus centros de control, empezando con una gran dependencia de acciones manuales y tendiendo a la automatización. Para ello, se han adaptado o reemplazado los antiguos mecanismos de control por nuevos dispositivos informatizados dándoles conectividad IP y uniéndolos a redes de ordenadores para poder ser operados a distancia. Esta informatización también ha llevado a una interconexión entre las diversas infraestructuras que no existía previamente, pudiéndose enviar y recibir información de unas a las otras para ganar en coordinación y cooperación.

A parte de la interconectividad que se produce al informatizar las infraestructuras, también tenemos que considerar las dependencias naturales - o interdependencias en el caso de que haya dependencias recíprocas - que existen entre ellas. Éstas pueden deberse a una causa de tipo física, TIC, geográfica o lógica[1].

Una dependencia física es aquella que conecta dos infraestructuras porqué una necesita lo que suministra la otra para poder operar. Por ejemplo, necesitamos que la producción de energía eléctrica funcione correctamente para poder mantener activo el servicio de telecomunicaciones.

Una dependencia TIC se da en las infraestructuras que necesitan de la información transmitida por otra infraestructura a través de la infraestructura de telecomunicaciones. Este tipo de dependencia va en aumento debido a la extensa utilización de los sistemas de control industrial para la Supervisión, Control y Adquisición de Datos (SCADA). Además, los centros de control que gestionan las infraestructuras tienden a estar cada vez más alejados de las infraestructuras que controlan, con lo que la dependencia con las TIC es todavía más enfatizada.

Las dependencias geográficas se encuentran en esos puntos en que se sitúan próximamente varias infraestructuras críticas, por ejemplo un puente donde pasan líneas de comunicaciones o canalizaciones de agua. Una afectación en el puente podría provocar problemas tanto en las líneas de tráfico, como en las telecomunicaciones o el suministro de agua.

Finalmente, las dependencias lógicas son aquellas que existen entre dos infraestructuras y que no corresponden ni a dependencias de tipo físico, ni TIC, ni geográfico. En este caso un agente en una infraestructura depende de algún modo de un agente en otra, pero el vínculo entre los dos se establece por algún mecanismo que no corresponde a los anteriormente mencionados. Un ejemplo de este tipo es la dependencia que se produce entre las infraestructuras eléctricas y las financieras. Desde la privatización del mercado eléctrico se han hecho muchas inversiones financieras en este sector. Por consiguiente, se producen cuantiosas pérdidas financieras cuando hay afectaciones en el precio de la energía, del transporte, al aplicar nueva regulación, nuevos impuestos, etc. Igualmente, afectaciones en los mercados financieros que hagan desplazar las inversiones en el sector eléctrico pueden no hacer rentables algunas plantas generadoras y desproveer a la red eléctrica de potencial.

En relación a los tipos de dependencia, cabe mencionar la clasificación respecto al nivel de criticidad que tiene un fallo y la temporalidad que ese fallo conlleva. La interrupción de producción de energía eléctrica en una planta de mediano tamaño puede no afectar demasiado al sistema en un día de poco consumo, pero ese mismo fallo puede llevar a una caída en cadena del sistema en un día con un pico de consumo. Además, también hay que distinguir en el grado de acoplamiento que tienen las infraestructuras. Por ejemplo, el corte de suministro de gas a un generador de ciclo combinado probablemente tendrá consecuencias casi inmediatas en la generación de energía eléctrica. En cambio, en las centrales de carbón, al disponer normalmente de reservas, un corte en el suministro no tendrá una consecuencia hasta al cabo de varios meses.

La conjunción de las tres características anteriores: incorporación de sistemas TIC, dependencias y mayor interconexión entre infraestructuras es la base de funcionamiento de un sistema smart city. Una vez establecida la interconexión, el sistema smart city se encarga de analizar las dependencias entre las infraestructuras con los datos en tiempo real provenientes de sensores y otros dispositivos repartidos por la ciudad y así ofrecer herramientas para un mejor control y operación de los diferentes servicios. Por ejemplo, una red de sensores en las calles que monitoreen el tráfico podría enviar información detallada del número de vehículos circulando por determinadas vías al centro de control de tráfico. Varias operaciones de análisis y predicción podrían alertar de los atascos y de las vías más rápidas. Esta información sería enviada al centro de control de ambulancias que combinando estos datos con los sistemas de posicionamiento planificaría las unidades mejor situadas para atender una urgencia y las rutas a tomar. En un sistema altamente conectado el centro de control de emergencias también podría enviar información al de tráfico sobre las intervenciones necesarias, que introducido en la red semafórica podría seguir el recorrido de los servicios de emergencia y darles prioridad.

II-C. Los datos y la información en una smart city

Una de las particularidades más destacadas de las smart cities es la gran cantidad de datos que manejan provenientes de fuentes heterogéneas distribuidas geográficamente por el área urbana. Si listamos estas fuentes clasificándolas con el vínculo que tienen con la identidad y la privacidad de los ciudadanos tenemos:

- **Fuentes no personales** correspondientes a los dispositivos que registran datos que no tienen ningún vínculo estrecho con una persona en concreto. E.g., sensores de temperatura, de humedad, sonómetros.
- **Fuentes personales** son aquellas vinculadas a un usuario donde su identidad aparece directamente. El ciudadano puede haber dado la información activamente, e.g., la participación en una aplicación de denuncia ciudadana, o de forma inconsciente sin saber que su información acabaría dentro del sistema, e.g., un comentario en una red social que es analizado por el sistema.
- **Fuentes anónimas** ofrecen datos que provienen de los usuarios pero se han tratado previamente para enmascarar la información personal de éstos. E.g., cámaras de videovigilancia que ensombrecen rostros, un parquímetro realizando la lectura de la matrícula de un coche. Hay que tener en cuenta que dependiendo de cómo se haya hecho el tratamiento de datos se podría deducir información de algunos usuarios. E.g., datos de consumo eléctrico en agregaciones espaciales de unos pocos kilómetros cuadrados pueden no revelar información sobre habitantes de regiones densamente pobladas, pero puede no ser suficiente para áreas rurales.

Cada vez hay más dispositivos que pueden nutrir datos al sistema smart city y a esto hay que añadir el movimiento open data. Este movimiento promueve que cada vez haya más datos que se abran. En muchos casos, para respetar la privacidad del ciudadano, la información se presenta agregada o anonimizada. Así, este modelo de más datos y más abiertos ayuda a la transparencia de las administraciones y empresas de servicio público, es fuente de creación de nuevas aplicaciones y servicios, pero también añade incertidumbre sobre los datos que se ofrecen y la manera en que se ha hecho el tratamiento de datos privados.

III. Seguridad y privacidad

Sin tener en cuenta los problemas derivados de las numerosas interconexiones y dependencias inherentes a una smart city, este tipo de sistemas también afrontan los problemas de seguridad informática clásicos que afectan a los centros de datos y a los sistemas de comunicación: malware (virus, troyanos, gusanos, backdoors, spyware), bots, loggers, rootkits, ataques de denegación de servicio distribuidos (DDoS), falta de actualizaciones, etc. Se han destacado los gusanos y los DDoS como los más peligrosos para los servicios que ofrece la smart city en tiempo real y para las infraestructuras críticas, ya que tienen una afectación muy alta para el rendimiento de los sistemas[11]. Típicamente, la prevención contra todos estos ataques se ha hecho con la instalación de antivirus, firewalls, honeypots, sistemas de detección de intrusiones (IDS), la creación de políticas de seguridad, la actualización de los sistemas y la implementación de medidas de autenticación. Los problemas de falta de actualización son de especial relevancia en las infraestructuras críticas, donde precisamente por su criticidad se minimizan las actualizaciones para evitar daños[12]. También hay dificultades para la actualización, la aplicación de nuevas políticas de autenticación o la denegación de autorizaciones en los dispositivos repartidos por la ciudad y que no disponen de una plataforma común de control.

En el ámbito de la privacidad, los problemas tradicionales que conciernen a las smart cities afectan a las bases de datos, a la identidad de los usuarios y a las comunicaciones.

En el campo de las bases de datos hay algunos procedimientos propuestos dirigidos a mantener la privacidad de los usuarios[13]. Las técnicas de Statistical Disclosure Control (SDC) proponen añadir ruido o hacer agregaciones para preservar la privacidad pero a su vez manteniendo el valor informativo de los datos. Las técnicas de Private Information Retrieval (PIR) se basan en hacer consultas pidiendo más información de la necesaria para ocultar la información concreta que demandaba el usuario. Otras técnicas como el cloaking y el uso de pseudónimos se usan para ocultar la identidad de los usuarios concretos al acceder a servicios basados en la localización (LBS).

Para los problemas de privacidad en las comunicaciones, la criptografía avanzada y el control de acceso son los sistemas usados para la prevención de escuchas en la transmisión de datos y para evitar la conexión de nodos no autorizados en las redes con aparatos distribuidos en lugares de acceso público[14]. Sin embargo, el uso de técnicas criptográficas puede ser viable para dispositivos con alta capacidad de cómputo, como los contadores inteligentes, pero sensores y otros dispositivos más pequeños pueden no tener capacidad suficiente para realizar estas funciones.

III-A. Problemas abiertos

La complejidad de un sistema crece exponencialmente al añadir nuevos subsistemas, y el número de vulnerabilidades que añade al conjunto este nuevo subsistema es mayor que las que lo afectaban de forma aislada[14]. Estas vulnerabilidades pueden ser aprovechadas por hackers o terroristas no sólo para causar daño al sistema que tiene abierta esta vulnerabilidad, sino que pueden utilizarla como puerta de entrada al resto de subsistemas que conforman la smart city.

El primer problema aparece al aumentar las interconexiones entre servicios, empresas e infraestructuras, ya que incrementamos también las vías para la circulación de virus entre objetivos codiciados como son las infraestructuras críticas. El ejemplo de infección del gusano Stuxnet[15] que ha afectado a los sistemas de control industrial aprovechándose de vulnerabilidades en sistemas Windows nos muestra la fragilidad y el riesgo de implantar las TIC e interconectar este tipo de entornos que anteriormente tenían su seguridad basada básicamente en seguridad física para impedir el acceso. Cómo los

virus, los hackers también pueden utilizar las interconexiones para viajar entre sistemas y ganar control.

El segundo problema en un sistema smart city procede de las dependencias entre infraestructuras. Un fallo en uno de los nodos en la red de dependencias podría causar problemas en cascada a varias infraestructuras críticas. Para una mejor planificación y gestión de las dependencias se han propuesto soluciones en el campo de la simulación y el modelado, pero los productos que sirven para una gestión global de múltiples infraestructuras están todavía poco maduros[16]. Este tipo de problemas en cascada puede deberse tanto a la disrupción de uno de los subsistemas como a la generación de desinformación. Continuando con el ejemplo de la sección II-B, si un atacante produjera un colapso en uno de los colectores que recoge los datos de los sensores de tráfico de un cruce importante y a su vez provocara una pequeña incidencia circulatoria, no solamente estaría afectando a las lecturas de tráfico, sino que el servicio de ambulancias estaría basando su planificación en datos desactualizados.

En tercer lugar, la conexión entre el middleware de la smart city y el resto de plataformas y aplicaciones es un elemento estratégico para que una smart city tenga éxito. Esta conexión tiene que ser interoperable, estandarizada y a su vez contemplar los principios básicos de confidencialidad, integridad y autenticidad. Por lo tanto, las APIs que ofrece la plataforma tienen que soportar el uso de protocolos con encriptación como HTTPS, hecho que crea algunos problemas. Por ejemplo, la posibilidad de que un atacante use un dispositivo en la vía pública con una conexión encriptada para enviar un virus hasta el subsistema de la smart city que desencripta la conexión. En este caso, un firewall perimetral no podría descifrar el contenido enviado y por lo tanto no podría detectar el virus. Un segundo ejemplo recae en que algunos dispositivos por su poca capacidad no soportan conexiones encriptadas. Aceptar que también sean posibles este tipo de conexiones para dispositivos inocuos abre la puerta a que sean atacados y también a errores humanos como malas configuraciones en otros elementos más peligrosos.

Un cuarto problema aparece con el hecho de disponer de muchos servicios y fuentes de datos para la creación de nuevas aplicaciones. Esto es una ventaja, pero a su vez es un riesgo al no poder asegurar la disponibilidad de estos servicios. Por ejemplo, una aplicación para la visualización del servicio de autobuses donde aparezcan las líneas y las paradas de autobús en un mapa cerca de la zona donde estamos con el tiempo de espera para cada autobús podría necesitar la disponibilidad de un servidor de mapas, de un servicio de localización para indicar al usuario donde se encuentra y del servicio que da información de forma dinámica en tiempo real sobre el tiempo de espera de los autobuses. Un fallo en cualquiera de estos servicios llevaría a la aplicación a no funcionar debidamente o incluso a que fuera inservible.

Finalmente, en el ámbito de la privacidad, a pesar de las técnicas mencionadas en la sección III, en un contexto de open data con cuantiosas fuentes de información tanto en tiempo real como históricas, al publicar nuevos datos parece difícil poder asegurar que no podrán ser utilizados para inferir la identidad de los usuarios al aplicar alguna técnica de correlación en el futuro. Un ejemplo del uso de técnicas de este tipo lo ha llevado a cabo la ciudad de Nueva York[17]. Para no tener que pagar para deshacerse de los aceites usados, algunos restaurantes los vierten ilegalmente en las alcantarillas. Correlacionando datos públicos provenientes del sistema de alcantarillado, información de contaminación, de licencias de restaurantes, de compañías de recogida de residuos entre otras, el ayuntamiento pudo dibujar un mapa de probabilidades de los restaurantes que habían cometido los vertidos ilegales sin disponer de ningún dato inicial que indicara la identidad de los autores.

Para disminuir el tiempo en las intrusiones y en los ataques, se propone la implantación de soluciones capaces de aplicar reacciones activas donde automáticamente el sistema responda en un escenario de crisis para frenar una anomalía[18]. Estas soluciones son todavía poco comunes, ya que los sistemas actuales están basados en la activación de alertas para la solución semimanual de irregularidades.

IV. Conclusión

En este artículo hemos presentado un esquema genérico de smart city en el que se basan algunos de los productos implementados por empresas y ciudades. A partir de este esquema, hemos repasado cómo se están integrando las infraestructuras críticas a la smart city a base de informatizarlas e interconectarlas. A las dependencias naturales que tienen entre sí estas infraestructuras, se le añaden dependencias con las TIC, que llevan a todo el sistema a ser más vulnerable a ciberataques y a ser susceptible a fallos múltiples en cascada. Así, hemos repasado los problemas de seguridad informática que el modelo de smart city tiene asociado. Además, el paradigma de open data en el que se basa la publicación de muchos de los datos generados por la smart city contrae problemas de privacidad para los ciudadanos y las empresas de las que se extraen esos datos. Varios de estos problemas de seguridad y privacidad han sido resueltos para otros entornos, pero debido a las particularidades y características propias de una smart city, algunos de los problemas continúan abiertos en este contexto.

Agradecimientos

Este trabajo está financiado parcialmente por el Ministerio de Economía y Competitividad a través de los proyectos TIN2011-27076-C03-02 “CO-PRIVACY” y CONSOLIDER INGENIO 2010 CSD2007-0004 “ARES”; y por la Generalitat de Catalunya a través de la subvención de doctorado industrial ECO/2497/2013. Merecen un agradecimiento especial el Ayuntamiento de Barcelona, Cast Info y openTrends.

Referencias

[1] S. M. Rinaldi, J. P. Peerenboom, and T. K. Kelly, “Identifying, understanding, and analyzing critical infrastructure interdependencies,” *Control Systems, IEEE*, vol. 21, no. 6, pp. 11–25, 2001.

[2] “Living planit os,” http://living-planit.com/UOS_overview.htm, accessed: 2014-03-27.

[3] F. Gil-Castineira, E. Costa-Montenegro, F. J. Gonzalez-Castano, C. López-Bravo, T. Ojala, and R. Bose, "Experiences inside the ubiquitous oulu smart city," *Computer*, vol. 44, no. 6, pp. 48–55, 2011.
[4] Y. W. Lee and S. Rho, "U-city portal for smart ubiquitous middleware," in *Advanced Communication Technology (ICACT), 2010 The 12th International Conference on*, vol. 1. IEEE, 2010, pp. 609–613.
[5] M. Chen, "Towards smart city: M2m communications with software agent intelligence," *Multimedia Tools and Applications*, vol. 67, no. 1, pp. 167–178, 2013.
[6] "Citysdk," http://www.citysdk.eu/, accessed: 2014-03-27.
[7] "icity," http://www.icityproject.com/, accessed: 2014-03-27.
[8] "Sentilo," http://www.sentilo.io/wordpress/, accessed: 2014-03-27.
[9] "Open cities," http://opencities.net/, accessed: 2014-03-27.
[10] F. Ferraz, C. Sampaio, C. Ferraz, G. Alexandre, and A. Carvalho, "Towards a smart city security model exploring smart cities elements based on nowadays solutions," in *ICSEA 2013, The Eighth International Conference on Software Engineering Advances*, 2013, pp. 546–550.
[11] C.-W. Ten, G. Manimaran, and C.-C. Liu, "Cybersecurity for critical infrastructures: attack and defense modeling," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 40, no. 4, pp. 853–865, 2010.
[12] F. Daryabar, A. Dehghantanha, N. I. Udzir, N. F. B. M. Sani, and S. bin Shamsuddin, "Towards secure model for scada systems," in *Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), 2012 International Conference on*. IEEE, 2012, pp. 60–64.
[13] A. Martinez-Balleste, P. A. Pérez-Martínez, and A. Solanas, "The pursuit of citizens' privacy: a privacy-aware smart city is possible," *Communications Magazine, IEEE*, vol. 51, no. 6, 2013.
[14] A. Bartoli, J. Hernández-Serrano, M. Soriano, M. Dohler, A. Kountouris, and D. Barthel, "Security and privacy in your smart city," in *Proceedings of the Barcelona Smart Cities Congress*, 2011.
[15] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *Security & Privacy, IEEE*, vol. 9, no. 3, pp. 49–51, 2011.
[16] S. M. Rinaldi, "Modeling and simulating critical infrastructures and their interdependencies," in *System sciences, 2004. Proceedings of the 37th annual Hawaii international conference on*. IEEE, 2004, pp. 8–pp.
[17] T. G., "Why grease is the word in new york," http://www.ft.com/cms/s/2/a284331a-9751-11e2-a77c-00144feabdc0.html#axzz2P5yLBdjC, 2013, accessed: 2014-03-27.
[18] L. Cazorla, C. Alcaraz, and J. Lopez, "Towards automatic critical infrastructure protection through machine learning," in *Critical Information Infrastructures Security*. Springer, 2013, pp. 197–203.

Desarrollando una metodología de análisis de riesgos para que el sector asegurador pueda tasar los riesgos en las PYMES

Antonio Santos-Olmo
Departamento de I+D+i
SICAMAN NT
Tomelloso, España
Email: Asolmo@sicaman-nt.com

Luis Enrique Sánchez
Departamento Eléctrica
y Electrónica
Universidad de las Fuerzas
Armadas Latacunga, Ecuador
Email: luisenrique@sanchezcrespo.org

Eduardo Fernández-Medina, Mario Piattini
Grupos de Investigación
ALARCOS y GSyA
Universidad de Castilla-La Mancha (UCLM)
Ciudad Real, España
Email: Eduardo.FdezMedina, Mario.Piattini@uclm.es

***Resumen*—En una sociedad gobernada por la información, las empresas y en particular las PYMES, dependen cada vez más de la capacidad de poder asegurar la información, no solo internamente, sino con terceros que estén dispuestos a establecer pólizas de seguros sobre la información. Pero cuando estamos hablando de activos intangibles, las aseguradoras se enfrentan a la problemática de que no existen metodologías de Análisis de Riesgos adecuadas que permitan tasar y garantizar la información de forma objetiva. En este artículo, presentamos la base de una nueva metodología que tiene como objetivo dar solución a las problemáticas presentadas por las empresas y las aseguradoras, permitiendo realizar un análisis de riesgo con menor grado de incertidumbre que los existentes en la actualidad.**

***Palabras clave*—PYMES; Analisis de riesgos; Tasación de activos; Aseguradoras**

I. Introducción

Para las empresas, es muy importante implantar controles de seguridad que les permitan conocer y controlar los riesgos a los que pueden estar sometidas [1], [2]. Pero la implantación de estos controles no es suficiente, siendo necesarios sistemas que gestionen la seguridad a lo largo del tiempo, de modo que les permitan reaccionar ágilmente ante nuevos riesgos, vulnerabilidades, amenazas, etc. [3]. Sin embargo, la mayor parte de las empresas tienen sistemas de seguridad caóticos creados sin unas guías adecuadas, sin documentación y con recursos insuficientes [4]. Los controles clásicos se muestran por sí solos insuficientes para dar unas mínimas garantías de seguridad. Por lo tanto, a pesar de que la realidad ha demostrado que para que las empresas puedan utilizar las tecnologías de la información y las comunicaciones con garantías es necesario disponer de guías, métricas y herramientas que les permitan conocer en cada momento su nivel de seguridad y las vulnerabilidades que aún no han sido cubiertas [5], el nivel de implantación con éxito de estos sistemas realmente es muy bajo. Este problema se acentúa especialmente en el caso de las PYMES, que cuentan con la limitación adicional de no tener recursos humanos y económicos suficientes para realizar una adecuada gestión [4]. Algunos autores [6], [7] sugieren la realización de un análisis de riesgos como parte fundamental en la PYME. Otros autores [8] proponen la necesidad de desarrollar un nuevo modelo de análisis de riesgos orientándolo directamente a las PYMES, considerando que el uso de técnicas de análisis y gestión de riesgos, así como el papel de terceros (Ej: aseguradoras), es necesario para poder garantizar la seguridad del sistema de información de las PYMES Como tal, toma especial relevancia la necesidad de obtener nuevas metodologías y modelos de análisis y gestión del riesgo, que permitan adaptarse a las PYMES, con el objetivo de eliminar (o al menos reducir) los inconvenientes y ayudar a estas sociedades a evaluar los riesgos a los que sus activos están expuestos y a establecer los controles de seguridad adecuados, permitiendo a su vez que esa evaluación de riesgos sea lo suficientemente objetiva, como para ser aceptada por terceros. De esta manera, el objetivo principal de este artículo es mostrar el framework que se está desarrollando con el objetivo de poder obtener una metodología de análisis de riesgos que dé solución a los problemas detectados en las investigaciones previas [9]. El artículo continúa en la sección II, describiendo brevemente el objetivo de la metodología y la problemática que pretende solucionar. En la sección III se presentan brevemente las propuestas de framework de la metodología. Finalmente, en la sección IV concluimos indicando cuál será el trabajo que desarrollaremos en el futuro.

II. Estado del arte

MARISMA (Metodología para Análisis de Riesgos SIstemático basado en Modelos Asociativos inteligentes y cuantificables) es la metodología que se está desarrollando con el fin de permitir la tasación objetiva de un sistema de información y la generación de un análisis de riesgo objetivo que tenga en cuenta aspectos asociativos y jerárquicos y sea de bajo coste en su generación y mantenimiento. Antes de iniciar la elaboración de MARISMA, se realizo una revisión sistemática siguiendo el método científico, que fue mostrada en la anterior edición de la RECSI [9], y de la que entre otros resultados se pueden destacar las siguientes conclusiones:

1. La mayor carencia detectada en las metodologías actuales es el elevado nivel de aspectos subjetivos que deben ser establecidos y que invalida los resultados obtenidos, o por lo menos limita su uso [9]. A lo largo de nuestra experiencia hemos comprobado que la elaboración de un análisis de riesgos por parte de dos consultores, sobre la misma compañía, utilizando la misma metodología y con los mismos interlocutores, puede dar dos resultados completamente diferentes, al entrar en juego muchos aspectos subjetivos que deben ser valorados según la experiencia y el criterio de los consultores. Esta problemática hace que los resultados obtenidos en un análisis de riesgos sean parcialmente útiles para la propia compañía, pero totalmente inútiles cuando hablamos de terceras partes (compañías asociadas, proveedores, clientes, aseguradoras).
2. La segunda carencia detectada en las metodologías actuales, es que actualmente las metodologías consideran que las empresas y los activos están aislados. En base a nuestra experiencia de auditoría y certificación con la norma ISO27001 [10], nos hemos dado cuenta que uno de los mayores puntos de polémica es la definición del .ªlcance a certificar", ya que obliga a establecer una frontera clara de qué activos están dentro del alcance y cuáles no. Estos aspectos de asociatividad y jerarquía deben ser contemplados en un análisis de riesgos para que los resultados tengan un valor real y adecuado.
3. La tercera carencia detectada es la falta de un sistema de tasación monetaria adecuado para los activos de información. En reuniones mantenidas con aseguradoras, se llegó a la conclusión de que, en la era del conocimiento, una de las pocas cosas que todavía no podían asegurarse y que suponían un mayor riesgo para las compañías eran los sistemas de información, y que aunque cada vez más compañías solicitaban el poder asegurar y tasar el sistema de información, las compañías no habían localizado ningún mecanismo objetivo que les permitiera asegurar una compañía con las garantías necesarias.
4. La cuarta carencia detectada es que la metodología que se construya debe adaptarse a las características requeridas por las PYMES, que principalmente exigen un bajo coste de recurso, tanto económicos, como de tiempo y personal.
5. Finalmente, la quinta y última carencia detectada es que actualmente las empresas desconocen las inter-relaciones de sus activos con sus clientes y proveedores, carecen de ese grafo, lo que hace que les sea difícil entender muchas veces los riesgos que asumen. Aquí introducimos un nuevo concepto que creemos que puede llegar a solucionar ese problema y que es el concepto de Red Social Empresarial.ªplicada al control y la gestión de las inter-relaciones entre los activos de las compañías derivados de sus estructuras empresarial, o de la aprobación de un proyecto.

La metodología propuesta resolverá todas estas carencias detectadas durante la investigación, buscando que los resultados obtenidos sean no sólo validos desde el punto de vista científico, sino que tengan una aplicación directa a las empresas objetivo de la investigación.

III. FRAMEWORK MARISMA

El principal objetivo de esta investigación es el desarrollo de un marco de trabajo metodológico que permita realizar análisis de riesgos con el menor grado de incertidumbre, que sean válidos para las PYMES, que sean dinámicos, controlen aspectos asociativos y jerárquicos y permitan la tasación económica y objetiva de los Sistemas de Información de una compañía.

De cara a hacer posible la obtención de una valoración económica objetiva de un sistema de información y de los riesgos a los que están sometidos estos activos, con el menor grado de incertidumbre, con el objetivo de permitir a una compañía aseguradora poder realizar un seguro del mismo, o permitir conocer a un tercero los riesgos que asume al ceder un activo o colaborar con la compañía, planteamos la necesidad de desarrollar un marco metodológico que permita realizar este proceso.

El marco metodológico estará formado por tres componentes:

- *MI*: Contendrá el modelo de información, y estará formado por las ontologías y las bases de conocimiento del marco metodológico.
- *I*: Contendrá todas las métricas que nos permitirán las tasaciones económicas objetivas de los activos, y las reducciones del nivel de incertidumbre en la elaboración del análisis de riesgos.
- *M*: Contendrá la propia metodología de tasación y análisis y gestión del riesgo.

En las siguientes sub-secciones se irán detallando los principales elementos y características del marco de trabajo que se está desarrollando.

Modelo de Información - MARISMA.MI

La primera parte del marco metodológico que proponemos, contendrá un modelo de información que recoge todos los conceptos relacionados con la metodología que se pretende desarrollar. Estará formada por un conjunto de ontologías y una base de conocimiento, que nos permitirá reutilizar el conocimiento adquirido en diferentes implantaciones, y que estará basada en las investigaciones realizadas por [11], [12], entre otras.

Para el desarrollo de estas ontologías, debemos ser capaces de analizar las tres dimensiones del problema:

- *Conceptos relacionados con el campo de la tasación de activos (TA)*: para abarcar este dominio del problema, analizaremos otras investigaciones y estándares existentes. Las investigaciones realizadas hasta el momento han concluido que existen muy pocos estudios y estándares relacionados con la materia [13], [14].

- *Conceptos relacionados con el campo de la seguridad (S)*: para abarcar este dominio del problema, utilizaremos los principales estándares relacionados con la gestión de la seguridad de Sistemas de Información, en especial los relacionados con el análisis y gestión de riesgos (ISO27001, ISO27002, ISO27005, MAGERIT, OCTAVE, ...) [10], [15]–[24] y orientados en especial a disminuir el nivel de incertidumbre de la generación de un análisis de riesgos .
- *Conceptos relacionados con la interrelación de compañías (asociatividad y jerarquía) (AJ)*: para abarcar este dominio del problema, y ante la ausencia de estándares oficiales, utilizaremos los estudios obtenidos durante la revisión sistemática, que serán complementados con los resultados prácticos obtenidos de aplicar la investigación en caso reales mediante el método científico "investigación en acción".

El conjunto resultante de analizar estos tres dominios sobre un campo común como son los sistemas de información, dará lugar a un conjunto de ontologías que podremos aplicar sobre la metodología que estamos desarrollando.

Indicadores - MARISMA.I

La segunda etapa para el desarrollo de nuestra metodología se está centrando en el estudio y desarrollo de un conjunto de indicadores, reglas de negocio y métricas vinculadas a los procesos seguridad de los sistemas de información.

Uno de los objetivos de esta fase es facilitar que pueda determinarse de forma semiautomática la valoración (tanto monetaria como en cuanto a importancia dentro de la empresa) de los activos del sistema de información.

Una vez que hemos desarrollado la primera fase del marco de trabajo y obtenida una ontología, ésta se utilizará entre otras cosas para obtener reglas del sistema de tasación. Por último, estas reglas se utilizarán para aplicar factores derivados de las posibles relaciones de cada activo, amenaza y vulnerabilidad en cuanto a la jerarquía y asociatividad de la compañía dentro de su entorno, buscando siempre reducir el nivel de incertidumbre.

El objetivo último perseguido en esta fase es ser capaces de localizar y desarrollar indicadores y métricas que nos permitan calcular de forma semi-automática los valores de los activos y el nivel de riesgo al que están expuestos, reduciendo el nivel de incertidumbre en la elaboración del análisis de riesgos. De esta forma, esta parte de la investigación permitirá la consecución completa de los siguientes objetivos: i) Diseñar métricas para la valoración y tasación de activos de información; ii) Diseñar métricas para la valoración de las amenazas; iii) Diseñar métricas para la valoración de activos de información en base a criterios de riesgo; iv) Diseñar métricas para la valoración de controles de seguridad en base a estándares existentes y para calcular la probabilidad de ocurrencia de una vulnerabilidad.

Metodología - MARISMA.M

La tercera parte del marco de trabajo que estamos desarrollando contiene la metodología que se aplicará para la tasación objetiva de un sistema de información y la generación de un análisis de riesgo objetivo que tenga en cuenta aspectos asociativos y jerárquicos, reutilización del conocimiento, dinamismo, y que sea válida para las PYMES.

La metodología MARISMA está constituida por los siguientes artefactos:

- *Sistema de Tasación de Activos (STA)*: Permite, a partir de la lista de activos de la compañía, obtener una tasación económica de los mismos. Esta tasación se realizará en base a criterios totalmente objetivos, de forma que el valor de los activos no varíe si dos consultores diferentes realizan la tasación sobre los mismos activos. La tasación tendrá en cuenta también que pueden actuar sobre el valor de un activo dos tipos de factores: i) Factores jerárquicos: Por ejemplo, en el caso de una empresa filial, es posible que un determinado activo no le pertenezca, sino que sea propiedad de la matriz. O que la matriz deje ese activo a la filial mediante un leasing, con lo que sólo poseerá un porcentaje del activo; ii) Factores asociativos: Por ejemplo, un producto del cual la compañía se encargue de desarrollar el software, siendo incorporado el hardware por otra compañía asociada. En este caso, el valor del producto tasable para la compañía será sólo el correspondiente a la parte software del mismo. Este proceso está formado por cinco tareas: T1 – Lista de activos de la compañía; T2 – Rellenar el conjunto de propiedades de los activos; T3 – Calcular el valor total del activo; T4 – Aplicar factores asociativos y jerárquicos; T5 – Calcular el valor del activo en la compañía.
- *Sistema de Valoración Objetivo de Amenazas (SVOA)*: Permite valorar en base a métricas objetivas la probabilidad de ocurrencia de cada posible amenaza que puede afectar a cada uno de los activos de la compañía. En este sistema será básica la Base de Conocimiento que se va alimentando de cada nueva implantación, de forma que se pueda calcular automáticamente la probabilidad de ocurrencia de una amenaza en función de la calculada previamente para otra compañía con similares características. Por ejemplo, en función del ámbito geográfico. Los valores calculados también se verán afectados por la aplicación de factores jerárquicos y asociativos. Este proceso está formado por tres tareas: T1 – Pedir características de la compañía; T2 – Pedir factores asociativos y jerárquicos; T3 – Calcular el nivel de amenaza de la compañía.
- *Sistema de Medición Objetiva de Vulnerabilidades (SMOV)*: Permite valorar mediante métricas objetivas la probabilidad de que una vulnerabilidad pueda ser explotada para una compañía. Este sistema trabaja como parte fundamental una ontología de vulnerabilidades, para cada una de las cuales se calculará la probabilidad de ocurrencia. Este valor se calculará en función del los niveles de cobertura de los controles implantados en la compañía. De esta forma, el sistema trabajará sobre la base de un listado de controles. Para la primera

versión de la metodología se empleará el listado de controles de seguridad de la Norma ISO 27001 [10]. Los valores calculados también se verán afectados por la aplicación de factores jerárquicos y asociativos. Este proceso está formado por cinco tareas: T1 – Calcular el nivel de cobertura de los controles; T2 – Lista de vulnerabilidades; T3 – Probabilidad de ocurrencia de la vulnerabilidad; T4 – Aplicar factores asociativos y jerárquicos; T5 – Calcular el valor del activo en la compañía.

- *Sistema de Valoración de Activos Objetivo (SVAO)*: Permite dar un valor, de forma cuantitativa y objetiva, a cada uno de los activos de la compañía sobre la base de los principales criterios de riesgo (Confidencialidad, Integridad, Disponibilidad y Legalidad). Para ello se emplearán métricas que tomen como base estos criterios de riesgo. Los valores calculados también se verán afectados por la aplicación de factores jerárquicos y asociativos. Este proceso está formado por cinco tareas: T1 – Lista de activos de la compañía; T2 – Rellenar el conjunto de propiedades de los activos; T3 – Calcular el valor total del activo; T4 – Aplicar factores asociativos y jerárquicos; T5 – Calcular el valor del activo en la compañía.

En función de las valoraciones obtenidas por los sistemas SMOV (Probabilidad de ocurrencia de vulnerabilidades) y SVAO (Valoración de activos en base a criterios de riesgo), podemos obtener un valor de riesgo objetivo para cada uno de los activos de la compañía. Para realizar esto, nos estamos basando en las investigaciones de Feng [25] sobre generalización de la teoría Bayesiana de probabilidad subjetiva, las del modelo híbrido (probabilístico y posibilístico) de Carlsson [26], y métodos de inferencia difusa (fuzzy inference) para desarrollar modelos inteligentes de evaluación de riesgos en línea (intelligent online risk assessment models) propuestos por Abraham [27], entre otras [28]–[36]. Todas ellas orientadas a disminuir el grado de incertidumbre en la generación del análisis de riesgos.

Una vez calculado un valor de riesgo objetivo para cada activo, se podría utilizar como base para el cálculo del seguro del Sistema de Información de la compañía, ya que contamos también con la valoración económica objetiva de cada activo calculada previamente en el sistema STA. Como comentamos anteriormente, para la valoración económica de los activos nos estamos basando en las investigaciones de Lambrinoudakis [13].

Como hemos visto, los factores jerárquicos y asociativos se aplican a todos y cada uno de los sistemas que conforman el núcleo de la metodología. Asimismo, para el diseño y aplicación de la misma es necesario contar con una tercer factor: La necesidad de que la metodología sea dinámica, de forma que si hay algún cambio en el sistema (Por ejemplo, añadir un nuevo activo o un control que originalmente no se aplicaba) se puedan recalcular los valores de riesgo y tasación de una forma automática y ágil. Para definir estos aspectos nos estamos basando en las investigaciones de [26], [32], [37]–[39].

IV. Conclusiones y trabajo futuro

En este trabajo se ha propuesto MARISMA, un marco de trabajo que permite la tasación objetiva de un sistema de información y la generación de un análisis de riesgos objetivo que tenga en cuenta aspectos asociativos y jerárquicos y sea de bajo coste en su generación y mantenimiento.

Durante la investigación, se han estudiado las principales metodologías existentes en el mercado relacionadas con la generación de análisis de riesgos y se ha realizado una revisión sistemática de los diferentes modelos y metodologías para el análisis y gestión de riesgos, con el objetivo de estudiar las propuestas centradas en riesgos asociativos y jerárquicos orientadas a PYMES.

Como resultado de esta revisión se ha podido establecer la importancia que tiene la gestión y el análisis de los riesgos sobre la seguridad de los Sistemas de Información en el desempeño y evolución sostenible de las empresas, ya que constituye un requisito básico para alcanzar la misión y los objetivos organizacionales en un entorno altamente competitivo.

Además, se han realizado reuniones y entrevistas en empresas privadas y sectores como el asegurador, para establecer las necesidades reales de las empresas y terceros, de forma que la investigación tenga una clara aplicación práctica.

Se ha podido validar durante la investigación la problemática de aplicar las metodologías existentes en el caso de las PYMES, ya que éstas han sido concebidas para grandes empresas, siendo la aplicación de este tipo de metodologías y modelos difícil y costosa para las PYMES [40]–[44].

El problema principal de todos los modelos de análisis y gestión de riesgos existentes es que no están teniendo éxito a la hora de implantarse en PYMES, debido principalmente a que:

- Unos fueron desarrollados pensando en organizaciones grandes (Grandes estándares como CRAMM [23], ISO/IEC 27005 [17], MAGERIT [20], OCTAVE [45], NIST SP 800-39 [46], MEHARI [21] o COBIT [47]) y en las estructuras organizativas asociadas a éstas.
- Otros [37], [39], [48] han intentando simplificar el modelo para que pudiera ser apto para compañías con recursos limitados, pero son modelos incompletos que sólo afrontan parte del problema, o intentan aportar unas guías básicas de los pasos a realizar, pero sin entrar en cómo evaluar y gestionar realmente los riesgos de una forma en la que el propio personal técnico de la empresa se pueda involucrar. Además, la mayoría son modelos teóricos y están todavía en desarrollo.
- La mayoría de las propuestas no tienen en cuenta la necesidad de contemplar riesgos jerárquicos y asociativos, factores cruciales en la estructura y funcionamiento actual de las empresas (en el que cada vez tiene más peso el uso de sistemas en Cloud), sobre todo de las PYMES.
- No existen formas objetivas de realizar un análisis de riesgo, dejando gran parte de la responsabilidad a los consultores, de forma que los resultados no tienen validez para terceros.

- La valoración económica de los activos de información es subjetiva, al no existir formas objetivas de valorarlo.

De esta forma, creemos que la investigación propuesta es el inicio de una propuesta detallada y ambiciosa, ya que solucionará gran parte de la problemática existente con las metodologías actuales y tendrá una clara aplicación práctica.

Las ventajas de la investigación propuesta son claras; la posibilidad de poder tener mecanismos de tasación de sistemas de información y de análisis de riesgos que sean objetivos, con coste reducidos y que tengan en cuenta las interrelaciones de los activos supone un cambio radical en la forma de ver los análisis de riesgo, ya que estos se convierten en herramientas útiles para los terceros (ej: las aseguradoras) y posibilita que las compañías tengan mecanismos objetivos de comparación de los riesgos cuando contratan un proyecto a otra compañías.

Todos los estándares y propuestas para la evaluación y gestión de riesgos estudiados en este trabajo son muy importantes, y sus aportaciones serán tenidas en cuenta para el desarrollo de una metodología que incluya todas las características deseadas.

Agradecimientos

Esta investigación es parte del proyecto PROMETEO financiado por la Secretaría Nacional de Educación Superior, Ciencia, Tecnología e Innovación (SENESCYT) del Gobierno de Ecuador, y el proyecto SIGMA-CC (Ministerio de Economía y Competitividad y el Fondo Europeo de Desarrollo Regional FEDER, TIN2012-36904).

Referencias

[1] Kluge, D. Formal Information Security Standards in German Medium Enterprises. in CONISAR: The Conference on Information Systems Applied Research. 2008.

[2] Dhillon, G. and J. Backhouse, Information System Security Management in the New Millennium. Communications of the ACM, 2000. 43(7): p. 125-128.

[3] Barlette, Y. and V. Vladislav. Exploring the Suitability of IS Security Management Standards for SMEs. in Hawaii International Conference on System Sciences, Proceedings of the 41st Annual. 2008. Waikoloa, HI, USA.

[4] Wiander, T. and J. Holappa, Theoretical Framework of ISO 17799 Compliant. Information Security Management System Using Novel ASD Method., in Technical Report, V.T.R.C.o. Finland, Editor 2006.

[5] Wiander, T. Implementing the ISO/IEC 17799 standard in practice - experiences on audit phases. in AISC '08: Proceedings of the sixth Australasian conference on Information security. 2008. Wollongong, Australia.

[6] Michalson, L., Information security and the law: threats and how to manage them. Convergence, 2003. 4(3): p. 34-38.

[7] Volonino, L. and S. Robinson. Principles and Practice of Information Security. in 1 edition, Anderson, Natalie E. 2004. New Jersey, EEUU.

[8] Spinellis, D. and D. Gritzalis. nformation Security Best Practise Dissemination: The ISA-EUNET Approach. in WISE 1:First World Conference on Information Security Education. 1999.

[9] A., S.-O., et al. Revisión Sistemática de Metodologías y Modelos para el Análisis y Gestión de Riesgos Asociativos y Jerárquicos para PYMES. in XII Reunión Española sobre Criptología y Seguridad de la Información (RECSI12). 2012. Donostia, San Sebastián (España): Septiembre, 2012.

[10] ISO/IEC27001, ISO/IEC 27001, Information Technology - Security Techniques Information security management systemys - Requirements., 2013.

[11] Alhawari, S., et al., Knowledge-Based Risk Management framework for Information Technology project. International Journal of Information Management, 2012. 32(1): p. 50-65.

[12] Hewett, R. and R. Seker, A Risk Assessment Model of Embedded Software Systems. 29th Annual IEEE/NASA Software Engineering Workshop (SEW'05), 2005: p. 8.

[13] Lambrinoudakis, C., et al., A formal model for pricing information systems insurance contracts. Computer Standards amp; Interfaces, 2005. 27(5): p. 521-532.

[14] Stewart, T.A., Trying to grasp the intangible. Fortune, 1995: p. 91.

[15] ISO/IEC13335, ISO/IEC 13335, Information Technology - Security Techniques - Management of Information and Communications Technology Security, 2004.

[16] ISO/IEC27002, ISO/IEC 27002:2005, the international standard Code of Practice for Information Security Management (en desarrollo). 2007.

[17] ISO/IEC27005, ISO/IEC 27005. Information Technology - Security Techniques - Information Security Risk Management Standard, 2008.

[18] Stoneburner, G., A. Goguen, and A. Feringa. Risk Management Guide for Information Technology Systems, NIST SP 800-30. 2009.

[19] 4360:2004, A.N., Standars Australia and Standards New Zealand. Risk Management2004, Sydney, NSW.

[20] MageritV2, Methodology for Information Systems Risk Analysis and Management (MAGERIT version 2), 2006, Ministerio de Administraciones Públicas (Spain).

[21] MEHARI. Club de la Sécurité de l'Information Français. 2009; Available from: https://www.clusif.asso.fr/.

[22] OCTAVE. CERT - Software Engineering Institute, Carnegie Mellon. 2009; Available from: http://www.cert.org/octave/.

[23] CRAMM. Siemens Enterprise Communications Ltd. ÇRAMM toolkit". 2009; Available from: http://www.cramm.com/.

[24] [24]ISO/IEC27002, ISO/IEC 27002, Information Technology - Security Techniques - The international standard Code of Practice for Information Security Management., 2013.

[25] Feng, N. and M. Li, An information systems security risk assessment model under uncertain environment. Applied Soft Computing, 2011. 11(7): p. 4332-4340.

[26] Carlsson, C. and R. Fullér, Predictive Probabilistic and Possibilistic Models Used for Risk Assessment of SLAs in Grid Computing. IPMU 2010, Part II, CCIS 81, 2010: p. 747-757.

[27] Abraham, A., Nature Inspired Online Real Risk Assessment Models for Security Systems. EuroISI 2008, LNCS 5376, 2008.

[28] Chang, S.-I., et al., The development of audit detection risk assessment system: Using the fuzzy theory and audit risk model. Expert Systems with Applications, 2008. 35(3): p. 1053-1067.

[29] Wang, P., et al., A Fuzzy Decision Model of Risk Assessment Through Fuzzy Preference Relations with Users'Confidence-interval. IEEE Computer Society AINA'06, 2006.

[30] Deng, Y., et al., Risk analysis in a linguistic environment: A fuzzy evidential reasoning-based approach. Expert Systems with Applications, 2011. 38(12): p. 15438-15446.

[31] Ngai, E.W.T. and F.K.T. Wat, Fuzzy decision support system for risk analysis in e-commerce development. Decision Support Systems, 2005. 40(2): p. 235-255.

[32] Kumar, V., M. Schuhmacher, and M. García, Integrated Fuzzy Approach for System Modeling and Risk Assessment. MDAI 2006, LNAI 3885, 2006: p. 227 - 238.

[33] Lin, M., Q. Wang, and J. Li, Methodology of Quantitative Risk Assessment for Information System Security. CIS 2005, Part II, LNAI 3802, 2005: p. 526 - 531.

[34] Lo, C.-C. and W.-J. Chen, A hybrid information security risk assessment procedure considering interdependences between controls. Expert Systems with Applications, 2012. 39(1): p. 247-257.

[35] Patel, S.C., J.H. Graham, and P.A.S. Ralston, Quantitatively assessing the vulnerability of critical information systems: A new method for evaluating security enhancements. International Journal of Information Management, 2008. 28(6): p. 483-491.

[36] Salmeron, J.L. and C. Lopez, A multicriteria approach for risks assessment in ERP maintenance. Journal of Systems and Software, 2010. 83(10): p. 1941-1953.

[37] Nachtigal, S., E-business Information Systems Security Design Paradigm and Model. Royal Holloway, University of London, Technical Report, 2009: p. 347.

[38] Arikan, A.E., Development of a risk management decision support system for international construction projects. Middle East Technical University, 2005: p. 118.

[39] Ma, W.-M., Study on Architecture-Oriented Information Security Risk Assessment Model. ICCCI 2010, Part III, LNAI 6423, 2010: p. 18-226.

[40] Batista, J. and A. Figueiredo, SPI in very small team: a case with CMM. Software Process Improvement and Practice, 2000. 5(4): p. 243-250.
[41] Hareton, L. and Y. Terence, A Process Framework for Small Projects. Software Process Improvement and Practice, 2001. 6: p. 67-83.
[42] Calvo-Manzano, J.A., et al., Experiences in the Application of Software Process Improvement in SMES. Software Quality Journal., 2004. 10(3): p. 261-273.
[43] Tuffley, A., B. Grove, and M. G., SPICE For Small Organisations. Software Process Improvement and Practice, 2004. 9: p. 23-31.
[44] Mekelburg, D., Sustaining Best Practices: How Real-World Software Organizations Improve Quality Processes. Software Quality Professional, 2005. 7(3): p. 4-13.
[45] Alberts, C.J. and A.J. Dorofee, OCTAVE Criteria, Version 2.0, 2001.
[46] NIST, Security Metrics Guide for Information Technology Systems, 2004.
[47] COBITv4.0, Cobit Guidelines, Information Security Audit and Control Association, 2006.
[48] Abdullah, H., A Risk Analysis and Risk Management Methodology for Mitigating Wireless Local Area Networks Intrusion Security Risks. University of Pretoria, 2006: p. 219.

Arquitectura de Seguridad Multinivel: Una Guía para las Organizaciones Modernas

Robson de Oliveira Albuquerque[1,2], Fábio Buiati[1,2], Luis Javier García Villalba[1]
[1]Grupo de Análisis, Seguridad y Sistemas (GASS), Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases s/n, Ciudad Universitaria, 28040 Madrid, España
Email: {robson, fabio, javiergv}@fdi.ucm.es

[2]Faculdade de Tecnologia, Universidade de Brasília (UnB)
Curso de Engenharia de Redes de Comunicação, Departamento de Engenharia Elétrica
CEP: 70910-900 - Brasília -DF - Brasil
Email: {robson, fabio.buiati}@redes.unb.br

***Resumen*—La información puede considerarse como el activo más importante de cualquier organización moderna. Garantizar la seguridad de esta información implica preservar la confidencialidad, la integridad y la disponibilidad de la misma, tríada conocida como CIA en inglés. Este trabajo presenta una arquitectura de seguridad multinivel motivado por la necesidad de considerar la información desde diferentes puntos de vista con el fin de protegerla. Además, se sugiere una nueva clasificación de los elementos de información, operaciones, entidades y componentes que se pueden integrar para mostrar las distintas fuentes de riesgos al tratar con información sensible. Se muestra también una visión general de cómo se trata y se representa actualmente la información y por qué es tan difícil garantizar la seguridad en todos los aspectos del tratamiento de la información.**

Palabras clave*—Arquitectura, confianza, seguridad de la información.** ***(Architecture, trust, information security)

I. Introducción

La gestión de seguridad de la información es fundamental en cualquier organización. Aun así, son muy pocos los modelos formales que ayudan a proteger eficazmente la información. Una manera de tratar el problema de la seguridad de la información es gestionar los riesgos desde diferentes puntos de vista. Estos riesgos están asociados a fenómenos naturales, riesgos tecnológicos y riesgos humanos [4]. Teniendo en cuenta estos aspectos, este trabajo propone una arquitectura multinivel para la gestión de riesgos de seguridad en las organizaciones modernas. Este trabajo está organizado en 7 secciones, siendo la primera la presente introducción. La Sección II recoge los trabajos relacionados más representativos. La Sección III propone una arquitectura de seguridad multinivel. La Sección IV presenta un modelo de confianza para la arquitectura multinivel. Por último, la Sección V muestra las principales conclusiones que se extraen de este trabajo.

II. Trabajo Relacionado

Mucho se ha dicho sobre normativas y estándares en seguridad de la información y sobre la importancia de su uso. Las normas de seguridad sirven como una guía para el desarrollo de un sistema de gestión de seguridad de la información.

Normas como la BS7799 e ISO 27000 [5] son guías ampliamente reconocidas en el área de la seguridad de la información. Plataformas como ITIL y COBIT [6] son utilizadas también en la administración de las tecnologías de la información con el fin de guiar a las organizaciones a aumentar su productividad y, en algunos aspectos, ayudan a mantener la seguridad de la información en términos de organización y metodología [7].

Sin embargo, el cumplimiento de las normas no garantiza en absoluto la seguridad. Para hacer frente a la seguridad de la información se requiere ir más allá del cumplimiento de normas o de mejores prácticas.

Respecto a las arquitecturas de seguridad de la información, el Zero Trust Model for Cybersecurity [8] sostiene un mensaje muy claro: dejar de confiar en los paquetes de datos como si fuesen personas. La idea subyacente es que el concepto de redes internas y externas debe cambiarse porque uno asume que todo el tráfico no es de confianza. Zero Trust viene a decir que los datos internos deben ser protegidos contra abusos procedentes de la red interna y que los datos externos deben ser protegidos en las redes públicas.

[9] señala que existe una necesidad de mejorar la seguridad de la información a nivel administrativo y organizacional. Por su parte, [11] [10] advierten de un cambio en la manera de cómo las personan se relacionan con la seguridad de la información, convirtiéndose además en el centro del problema.

Con el fin de proteger la información, es muy importante entender la forma en que se trata en el mundo digital. Desde la perspectiva del usuario, la información puede ser un texto, una imagen o una combinación de ambos. Internet redefinió la forma de representarla y de recuperarla [12]. La representación de la información requiere de complementos estructurales o semánticas adicionales, que transforman los datos en algo significativo para los seres humanos.

Considerando todo lo expuesto anteriormente, las arquitecturas de seguridad actuales no logran gestionar los riesgos, las políticas, las personas y los activos de forma correcta. Para intentar paliar esta carencia, este trabajo propone una arquitectura de seguridad de información multinivel que trata

de conectar todas las piezas entre sí respecto a la seguridad de la información. La especificación del modelo en niveles es importante para ver cómo todos los elementos de la arquitectura de seguridad interactúan.

III. Arquitectura de Seguridad de la Información

La forma en la que vemos la seguridad está basada en una arquitectura multinivel. En este enfoque cada elemento es una pieza del rompecabezas que debe estar bien conectada, de forma que la seguridad de información pueda ser vista como un todo indivisible. La Figura 1 ilustra la arquitectura de seguridad de la información propuesta con sus niveles.

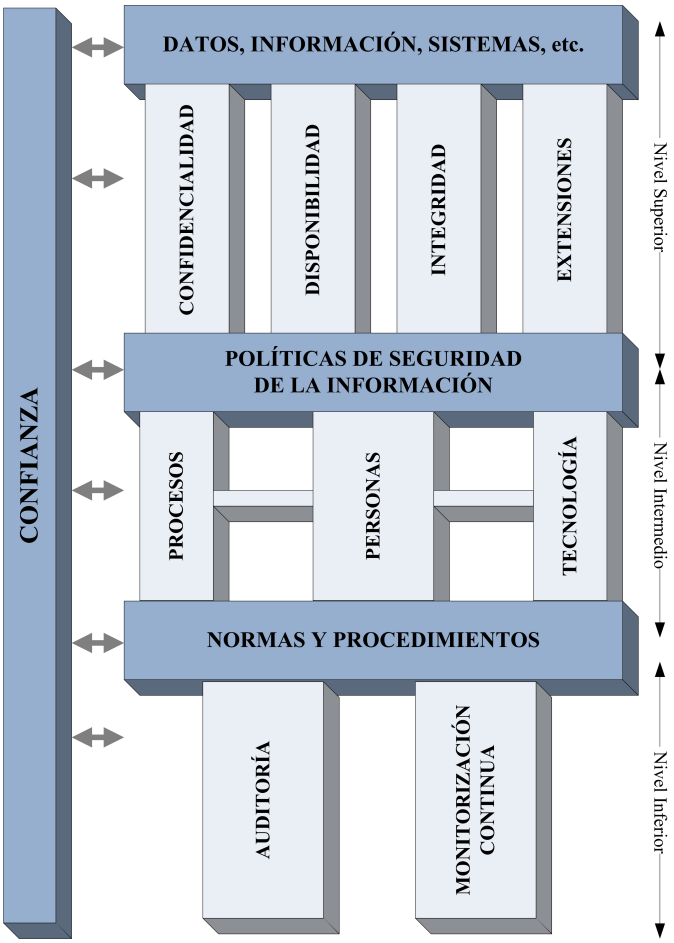


Figura 1. Arquitectura de Seguridad Multinivel

III-A. Nivel Superior

El nivel superior es la base para empezar a pensar en la seguridad de información de cualquier organización. Sin la adecuada comprensión de lo que son los datos, la información, los activos de información, etc., no hay cómo hablar de seguridad de la información, simplemente porque uno no sabe qué hay que proteger. Es importante señalar que el enfoque de "proteger todo" no es eficaz y es, además, bastante costoso.

En general, en este nivel es donde se localizan los datos importantes o con valor para las organizaciones o las personas. Considerando la importancia que tienen los datos actualmente, una gran cantidad de información se puede recuperar a partir de los datos y los sistemas de información. Utilizando herramientas y técnicas adecuadas, es posible crear además nuevos conocimientos a partir de los datos que, a simple vista, no parecen tener ningún sentido.

Cuando se trata de activos de información es muy importante que estos sean identificados y etiquetados, y la relación con la información debe ser claramente entendida por la organización.

Las redes de comunicación conectan los datos, la información y sus activos para que cualquier persona con acceso autorizado pueda explorarlas. Quién controla (personas) o cómo se controla (proceso, hardware o software) la red es lo que la hace peligrosa o no. Así que la creación de perímetros de redes, políticas y otros mecanismos de defensa sigue siendo una forma de controlar lo que entra y sale de la red. El uso de estos mecanismos es clave para entender lo que sucede en la transmisión dentro de los sistemas de información.

También en este nivel la seguridad tiene como foco salvaguardar la confidencialidad, la integridad y la disponibilidad de la información, debiendo aplicarse de forma efectiva en toda la cadena. La confidencialidad se refiere a la limitación de acceso a la información y a la divulgación a los usuarios autorizados. La integridad se refiere a la fiabilidad de los recursos de información, es decir, que los datos no han sido modificados inapropiadamente, ya sea por accidente o deliberadamente. Por último, la disponibilidad se refiere a la disponibilidad de los recursos de información.

Las extensiones de seguridad de la información son nuevos atributos o propiedades que protegen la información y los sistemas, pero no se limitan a ellos. La autenticación, el control de acceso, el no repudio, la privacidad, el anonimato y la autorización son servicios que se caracterizan como extensiones de seguridad.

III-B. Nivel Intermedio

Siguiendo un recorrido descendente nos encontramos con este nivel que es la parte de la arquitectura que nos ayudará a definir cuestiones tales quién, cómo, por qué y qué tecnologías pueden utilizarse para garantizar la seguridad de la información en el nivel superior. Los siguientes ítems son analizados: políticas de seguridad, procesos, personas y tecnología.

Una política de seguridad de la información es un documento de alto nivel que describe los requisitos o reglas que se deben cumplir para garantizar la seguridad de la información en una organización. En general, esta política es muy específica y cubre una única organización. La política de seguridad también está relacionada con los problemas de gestión y de control de la información, una vez que la protección de la misma está directamente relacionada con la cultura de la organización.

La política de seguridad debe explicar la necesidad de la seguridad de la información para todos los usuarios dentro de la organización y complementar los objetivos de la organización,

siendo necesario que esté alineada con el plan estratégico de la organización [13].

En la seguridad de la información los procesos son una manera formal de identificar, medir, gestionar y controlar los riesgos relacionados con la información o su valor para la organización. Los procesos incluyen mecanismos formales e informales (grandes o pequeños, simples o complejos, ...) para hacer las cosas y proporcionar un vínculo vital para todas las interconexiones dinámicas.

Las personas son el principal bloque del rompecabezas y representan el recurso humano. En general, una persona diseña e implementa cada parte de la política de seguridad, crea y mantiene los procesos, los activos de información, la tecnología utilizada, etc. Los problemas de seguridad afectan a las personas, sus relaciones, sus valores y sus comportamientos. Cuando se trabaja con seguridad de la información es importante hacer frente a puntos como las estrategias relacionadas con la contratación, el acceso, las responsabilidades, la formación, el despido, las sanciones y todo lo que sea importante abordar para ayudar a mantener la estrategia de seguridad de la información de la organización.

La tecnología es el elemento del rompecabezas constituido por un conjunto de sistemas de información, aplicaciones, herramientas, infraestructura y mecanismos de defensa que la organización utiliza para llevar a cabo su misión de proteger la información. Los elementos tecnológicos son susceptibles a frecuentes cambios y actualizaciones y pueden hacerse obsoletos rápidamente. La tecnología puede ser la parte fundamental de una infraestructura de la organización. La tecnología se usa también para resolver las amenazas de seguridad y los riesgos.

Es muy importante tener en cuenta que la tecnología por sí misma no hace nada. Debe ser vista como una parte de un sistema complejo que tiene necesidades específicas para proteger lo que es valioso en la organización. Además, la tecnología debe trabajar conjuntamente con personas y procesos completando un ciclo, todos ellos guiados por la política de seguridad de la información de la organización.

III-C. Nivel Inferior

Este nivel trata de las actividades diarias y las medidas que se deben adoptar en caso de un problema específico. Las prácticas de seguridad son guías para mantener la información segura. Sin embargo, las normas, procedimientos de monitorización y auditoría dan a los administradores las herramientas necesarias para ayudarles a mantener la información, los activos, las redes, los sistemas, etc., más seguros. Los siguientes ítems son analizados: normativas de seguridad, auditoría y monitorización continua.

Básicamente, una normativa define cómo deberían ser las cosas y cómo hay que valorarlas. También tiene que ver con la forma de clasificar las acciones en correctas o equivocadas. Las normativas son primordiales para la priorización de los objetivos y para definir cómo se deben hacer las cosas.

La auditoría de la seguridad de la información es un proceso que determina la valoración cualitativa y cuantitativa del estado actual del sistema analizado según criterios específicos de seguridad de la información. El proceso de auditoría es clave para encontrar riesgos, fallos técnicos, políticas, procedimientos y problemas normativos en una organización. Hay que tener en cuenta que la auditoría es un proceso que nunca termina. Cuando se realiza la auditoría, uno debe estar preparado para abarcar temas desde seguridad física de los centros de datos hasta la seguridad lógica, incluyendo los perímetros de red, la configuración del sistema y los sistemas de información.

Otra de las tareas realizadas en este nivel es la monitorización continua. Se trata de una actividad de mantenimiento de los conocimientos de seguridad de la información, vulnerabilidades, amenazas y riesgos asociados [14]. Es un punto clave de apoyo a la toma de decisiones relativas a la gestión de riesgos de una organización.

La monitorización continua se inicia definiendo qué, cómo, por qué y cuándo monitorizar los activos de información o cualquier parte de la arquitectura. Se apoya en tecnología, procesos, procedimientos, entornos operativos y personas. También ayuda en el establecimiento de prioridades y gestiona el riesgo de forma coherente en toda la organización.

IV. Confianza

Desde el punto de vista de la seguridad de la información, la confianza puede tener un valor de cero o de uno. Uno confía o no en sus sistemas de información, redes, activos, etc. El “tal vez” debe evitarse a toda costa. Por lo general, la confianza se adquiere mediante la observación empírica, por prueba formal de los sistemas, etc. [15].

La confianza y la seguridad están estrechamente relacionadas [15]. Si se consideran los objetivos de seguridad, está claro que los aspectos de confianza están conectados con la seguridad ya que mantener la información segura depende de las personas, las extensiones de seguridad (autenticación, autorización, control de acceso, no repudio, etc.).

Considerando lo anteriormente expuesto, no se puede proteger la información sin ser capaz de comprender todo el ciclo de vida que tiene la información. Hay que tener en cuenta una visión detallada si se desea más seguridad en el sistema; uno debe ser capaz de representar, procesar y utilizar la información en un entorno donde las personas, la tecnología, los activos de información, el hardware, el software, etc., están conectados entre sí. Y, paralelamente, hay que tomar medidas de seguridad para garantizar su protección. Ahí es donde la arquitectura de seguridad de la información multinivel con confianza entra en escena porque sólo proteger una parte de la información se ha demostrado ineficaz, como se ha visto recientemente [1][2].

La confianza en general es parte del rompecabezas cuando hay un conocimiento suficiente de la información, los sistemas, la tecnología y los demás componentes que ayudan hacer afirmaciones como “totalmente seguro” o la información es segura porque se cumple alguna condición en particular. Esta arquitectura en niveles le permite a uno hacer frente a determinados componentes y aislar problemas relacionados con cada uno de ellos.

V. Conclusiones

La tarea de garantizar la seguridad de la información no es un fin en sí mismo; es un medio para lograr un fin [16]. Se trata también de un tema en constante evolución, debido a la creciente magnitud y complejidad de las amenazas de seguridad de la era digital. Como se observa en la actualidad, el campo de investigación de la seguridad de información es cada vez más importante porque el mundo está interconectado con redes de comunicación que se utilizan para la transmisión de información crítica y sensible.

En este trabajo se ha introducido una arquitectura de seguridad multinivel donde los elementos de seguridad de la información están interconectados siendo útiles para la gestión de riesgos en los diferentes niveles de la organización. De esta forma, la seguridad de la información puede ser vista como un todo.

Gobierno, organizaciones y empresas que consideran la gestión de seguridad de la información necesitan un enfoque sistemático para abordar de manera coherente la seguridad en cada nivel, disminuyendo así los riesgos de administración y mejorando la eficiencia de la gestión de la seguridad. Bajo esta perspectiva, la arquitectura de seguridad de la información en niveles puede ser utilizada como una guía para obtener mejores resultados en la protección de la información.

Agradecimientos

Los autores también agradecen el apoyo proporcionado por el Laboratorio de Tecnologías de Decisión de la Universidad de Brasilia (LATITUDE / UnB). Asimismo, Fábio quiere agradecer la financiación que le brinda el Programa Nacional de Post-Doctorado de Brasil (PNPD/CAPES). El Grupo de Investigación GASS agradece la infraestructura proporcionada por el Campus de Excelencia Internacional (CEI) Campus Moncloa - Clúster de Cambio Global y Nuevas Energías (y, más concretamente, el sistema EOLO como recurso de computación de alto rendimiento HPC - High Performance Computing), infraestructura financiada por el Ministerio de Educación, Cultura y Deporte (MECD) y por el Ministerio de Economía y Competitividad (MINECO).

Referencias

[1] G. Greenwald, E. MacAskill, L. Poitras, “Edward Snowden: the whistleblower behind the NSA surveillance revelations,” *The Guardian*, Vol. 9, 2013.

[2] J.-T. Richelson, “The Snowden Affair. Web Resource Documents the Latest Firestorm over the National Security Agency,” *National Security Archive Electronic Briefing Book*, No. 436, 2013.

[3] Gartner Press Release. “Gartner Says Cloud-Based Security Services Market to Reach 2.1 Billion dollars in 2013”, Stamford, Conn., 2013. Disponible en ttp://www.gartner.com/newsroom/id/2616115.

[4] B. Blakley, E. McDermott, D. Geer, “Information security is information risk management,” *ACM Proceedings of the Workshop on New security Paradigms*, pp. 97–104, 2001.

[5] M. Whitman, H. Mattord, “Management of Information Security,” *Cengage Learning, Fourth Edition*, 2013.

[6] R. Parvizi, F. Oghbaei, S. R. Khayami, “Using COBIT and ITIL frameworks to establish the alignment of business and IT organizations as one of the critical success factors in ERP implementation,” *5th Conference on Information and Knowledge Technology (IKT)*, 2013 pp. 274–278, 2013.

[7] Department of Communications, Information Technology and the Arts and the Trusted Information Sharing Network. “Secure Your Information: Information Security Principles for Enterprise Architecture,” *Report, Australia*, 2007.

[8] The National Institute of Science and Technology (NIST), “Developing a Framework to Improve Critical Infrastructure Cybersecurity. Submitted by Forrester Research. In Response to RFI# 130208119-3119-01”, 2013. Disponible en http://csrc.nist.gov/cyberframework/rfi_comments/040813_forrester_research.pdf.

[9] R.-M. Ahlfeldt, P. Spagnoletti, G. Sindre. “Improving the Information Security Model by using TFI”, *In the New Approaches for Security, Privacy and Trust in Complex Environments. IFIP International Federation for Information Processing*, Vol. 232, pp. 73–84, 2007.

[10] R. Blakley, A. Johnston, P. Lowry, Q. Hu, M. Warkentin, R. Baskerville, “Future directions for behavioral information security research,” *Computers & Security*, Volume 32, February 2013, pp. 90–101.

[11] S. Aurigemma, R. Panko, “A Composite Framework for Behavioral Compliance with Information Security Policies”, *In proceedings of the 45th Hawaii International Conference on System Sciences. IEEE Computer Society*, 2012.

[12] H. Chu, “Information Representation and Retrieval in the Digital Age”, *Information Today, Inc*, Second Edition, 2010.

[13] ISACA. “An Introduction to the Business Model for Information Security”, 2009, Disponible en http://www.isaca.org/Knowledge-Center/BMIS/Documents/IntrotoBMIS.pdf.

[14] The National Institute of Science and Technology (NIST), “Information Security Continuous Monitoring (ISCM) for Federal Information Systems and Organizations”, *NIST Special Publication*, pp. 800–137, 2011.

[15] P. Lamsal, “Understanding Trust and Security”, *Department of Computer Science. University of Helsinki, Finland*, 2001.

[16] T. Peltier, “Information security fundamentals,” *CRC Press*, 2013.

Hacia la seguridad criptográfica en sistemas DaaS

Rafael Álvarez
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
Email: ralvarez@dccia.ua.es

Juan Santonja
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
Email: jsl2@alu.ua.es

Antonio Zamora
Departamento de Ciencia de la Computación e Inteligencia Artificial
Universidad de Alicante
Email: zamora@dccia.ua.es

***Resumen*—El nuevo paradigma de computación en la nube posibilita la prestación de servicios por terceros. Entre ellos, se encuentra el de las bases de datos como servicio (*DaaS*) que permite externalizar la gestión y alojamiento del sistema de gestión de base de datos.**

Si bien esto puede resultar muy beneficioso (reducción de costes, gestión simplificada, etc.), plantea algunas dificultades respecto a la funcionalidad, el rendimiento y, en especial, la seguridad de dichos servicios.

En este trabajo se describen algunas de las propuestas de seguridad en sistemas *DaaS* existentes y se realiza un análisis de sus características principales, introduciendo un nuevo enfoque basado en tecnologías no exclusivamente relacionales (*NoSQL*) que presenta ventajas respecto a la escalabilidad y el rendimiento.

***Palabras clave*—Base de datos (database), Cloud Computing, Cifrado homomórfico (Homorphic encryption).**

I. Introducción

En la actualidad estamos asistiendo al rápido despliegue del modelo de computación llamado *cloud computing* (computación en la nube). Este paradigma de computación ofrece un nuevo modelo de prestación de servicios tecnológicos y de negocio. A medida que va creciendo la oferta de proveedores, también lo hace la diversidad de servicios que ofrecen, así tenemos servicios *SaaS* (software as a service), *IaaS* (Infrastructure as a Service), *PaaS* (Platform as a Service) y *DaaS* (Database as a Service). En *DaaS*, el modelo permite ofrecer bases de datos como servicio.

Es este último caso el que resulta más interesante dada la problemática relacionada tanto con la seguridad como con la funcionalidad cuando externalizamos una base de datos. En concreto, los sistemas de bases de datos relacionales son los más utilizados para modelar problemas reales y administrar datos dinámicamente.

Los sistemas de gestión de bases de datos relacionales (*RDBMS*, Relational Database Management Systems) son uno de los elementos esenciales en los sistemas de computación actuales, ya que permiten almacenar y administrar datos en forma de tablas, de forma razonablemente sencilla mediante la definición de relaciones entre tuplas y elementos estructurales que limitan la existencia de duplicados, permitiendo la unión y búsqueda de elementos dentro de las tablas de forma óptima.

Las bases de datos relacionales utilizan el lenguaje de consulta estructurado o *SQL* (Structured Query Language). Dicho lenguaje, basado en el álgebra relacional, permite realizar tanto consultas de información a la base de datos como de modificación de estructura.

Son muchos los *RDBMS* comerciales existentes en la actualidad, entre los más extendidos se encuentran *MySQL* (y *MariaDB*), *PostgreSQL*, *Oracle*, *DB2*, *INFORMIX* y *Microsoft SQL Server*. Todos ellos, junto a las características propias del sistema de gestión de datos, poseen controles de seguridad como la definición de roles de acceso, logs de auditoría o, en los más avanzados, sistemas de cifrado de la información.

Curino et al. [4] indican que el uso de sistemas *DaaS* en el *cloud* es interesante desde el punto de vista económico por dos motivos, fundamentalmente: el primero, relacionado con la reducción de consumo energético ya que los costes son menores cuando los recursos son compartidos por varios usuarios; el segundo, relacionado con los costes de gestión, tanto de licencias como gastos administrativos que son menores en sistemas compartidos.

Independientemente de los motivos económicos, existen una serie de factores (véase [5]) que incentivan el uso de servicios *DaaS*: la escalabilidad horizontal que permite que los recursos puedan ser ampliados casi sin límite en el *cloud*; la velocidad de despliegue de aplicaciones e infraestructura que es mucho más rápida en sistemas compartidos que en sistemas propios; la flexibilidad en la contratación de servicios específicos evitando los costes de aquellos elementos que no son necesarios y, por último, la mayor fiabilidad de los proveedores de *cloud* que disponen de sistemas redundantes de respaldo e infraestructura, mejorando la disponibilidad de los servicios.

Aun cuando son muchas las ventajas, existen una serie de inconvenientes (véase [5]) que deben considerarse a la hora de elegir un sistema *DaaS*, como son la velocidad, el rendimiento, los costes asociados a la gestión de grandes volúmenes de información (*Big Data*) y la pérdida de control sobre la información. Es este último aspecto, el de la seguridad, el eje fundamental del presente trabajo.

Al externalizar un sistema relacional y ubicarlo en el *cloud* la información queda expuesta al administrador del sistema o cualquier elemento con acceso directo a la base de datos. Una posible solución para proteger los datos, garantizando la confidencialidad de los mismos, consiste en cifrar la información; con el inconveniente de que al realizar dicha acción, algunas de las propiedades básicas de los sistemas relacionales no son viables. El reto consiste, pues, en construir un esquema de

cifrado de la base de datos que permita la viabilidad de dichas propiedades (véase [16]).

Entre esas propiedades básicas destacamos las siguientes operaciones:

1. Relaciones entre tablas: Las tablas se relacionan entre sí mediante claves.
   a) Clave Primaria: permite identificar un registro de forma única.
   b) Clave Ajena: aparece en una tabla haciendo referencia a la información de otra tabla.
2. Operación Insertar: Añadir registros a una tabla
3. Operación Consulta: Seleccionar registros dentro de una tabla
   a) Consulta de atributos alfanuméricos
   b) Consulta de intervalos
   c) Consulta de agregación
4. Operación Modificar: Actualización de datos de una tabla.
5. Operación Eliminar: Borrado de registros

Existen diversas soluciones a este problema propuestas en la literatura. En este trabajo se describen y analizan algunas de ellas y se propone una nueva alternativa basada el paradigma no exclusivamente relacional (*NoSQL*).

II. Sistemas DaaS relacionales

II-A. Sistemas con cifrado en descanso (at rest)

Muchos de los productos existentes basan su seguridad exclusivamente en el cifrado del almacenamiento, de esta forma los datos se descifran de forma transparente al ser accedidos y se cifran al ser guardados, permaneciendo en claro mientras perduren en memoria. A pesar de la eficiencia en el rendimiento, el principal inconveniente que plantea este modelo consiste en que el proveedor elige (y, por tanto, dispone de) las claves de cifrado.

Aunque este modelo viene motivado por exigencias normativas, no resulta adecuado en aquellos casos en los que el proveedor no deba tener acceso a los datos, como ocurre en arquitecturas de tipo *DaaS* con datos sensibles.

Este modelo es adecuado para la protección ante un eventual robo de medios físicos, como unidades de disco o cintas de seguridad, ya que impediría el acceso a la información en claro. Las operaciones que realiza el sistema relacional hacen uso de los datos en memoria, que se encuentran en claro.

Algunas de las soluciones más utilizadas para este modelo son *Oracle* [10] y *SQL Server* [9].

II-B. Esquema de cifrado homomórfico completo

Aun cuando el modelo no tiene una implementación práctica viable en la actualidad, la propuesta de Gentry [6] resolvería completamente el reto propuesto. Consiste en un sistema de cifrado homomórfico, capaz de soportar tanto las operaciones de suma como de producto.

El principal problema de esta propuesta radica, como se ha indicado, en que su implementación no es viable con los medios computacionales disponibles en la actualidad. Por ejemplo, el sistema homomórfico parcial, requiere para el sistema más pequeño (512 dimensiones) un ancho de palabra de 200.000 bits, lo cual muestra la magnitud del problema. La clave pública usada en el sistema totalmente homomórfico tiene un tamaño de 17 MB y necesita 2.4 segundos para generarse. El sistema mayor (32768 dimensiones) requiere dos horas para generar la clave y ocupa 2.3GB [2].

II-C. Propuesta de L.M.X. Rodríguez

El esquema de cifrado para bases de datos relacionales propuesto por Rodríguez (véase [16]) pretende conseguir dos objetivos: garantizar la confidencialidad de la información y permitir que las consultas puedan ser procesadas por un sistema de base de datos relacional convencional. La propuesta utiliza diferentes tipos de primitivas criptográficas: un cifrador por bloques (AES), un cifrador homomórfico (Paillier [11]) y un cifrador que preserva el orden (Boldyreva [3]). Cada uno permite solventar limitaciones distintas a la hora de realizar consultas relacionales sobre un sistema de bases de datos cifrado.

Cifrador por bloques (AES): dado que los cifradores por bloques en modo directo (*ECB*) son deterministas (obtienen siempre el mismo texto cifrado para el mismo texto en claro con igual clave), permiten realizar las consultas que involucren campos alfanuméricos tanto para búsqueda (*SELECT*) como condicionales (*WHERE*). En esta propuesta se ha seleccionado el algoritmo *AES* por su buen rendimiento y ser un estándar bien conocido.

Cifrador basado en homomorfismos (Paillier): en las bases de datos es habitual realizar operaciones de totales y subtotales. Para permitir que el servidor realice operaciones que impliquen sumas, en esta propuesta se emplea un criptosistema homomórfico bajo la suma, como el de Paillier [11], en el que el resultado del producto de dos textos cifrados es idéntico al cifrado de la suma de esos textos sin cifrar; de esta manera se pueden sumar los datos sin conocerlos en claro.

Cifrador que preserva el orden (Boldyreva): en este tipo de cifrador, el texto cifrado presenta la característica de preservar el orden numérico del texto en claro; posibilitando la realización de consultas que impliquen la evaluación de un intervalo de información, estableciendo límites tanto inferiores como superiores.

La propuesta de Rodríguez [16] consta de tres elementos esenciales: el proceso de cifrado, el modelo de almacenaje y el proceso de consulta de la información cifrada.

- El proceso de cifrado recibe como entrada un registro de texto en claro el cual es analizado para elegir el método de cifrado apropiado para cada campo: los campos numéricos se cifran simultáneamente mediante el cifrador basado en homomorfismos y el que preservan el orden mientras que los alfanuméricos se cifran con el cifrador por bloques.
- El modelo de almacenaje se ve condicionado por el proceso de cifrado, dado que por cada campo numérico en claro se generan dos campos cifrados. El acceso a

cada campo cifrado se realiza en función de la operación requerida en la consulta.

- El proceso de consulta de información cifrada requiere la traducción de dichas consultas. El cliente solicita la consulta en claro y se analiza para determinar qué tipo de algoritmo aplicar: para el caso de restricciones alfanuméricas se accede a los campos cifrados con AES, para las comparaciones numéricas se accede a los campos cifrados con Boldyreva y, finalmente, para los cálculos que impliquen sumas se accede a los campos cifrados con Paillier.

Las limitaciones identificadas en el desarrollo de esta propuesta incluyen la imposibilidad de la ejecución de determinadas consultas, como búsquedas textuales mediante expresiones regulares, multiplicaciones, divisiones, números en coma flotante y fechas.

II-D. CryptDB

En la propuesta de Popa, et al. (véase [12], [13], [14], [15]) se desarrolla un sistema para garantizar la confidencialidad de la información en base a la ejecución de consultas SQL sobre datos cifrados utilizando un conjunto de esquemas de cifrado eficientes.

Este sistema permite gestionar las claves de cifrado a nivel de usuario, de modo que en una tabla puede haber datos de varios usuarios y sólo el usuario autorizado es capaz de descifrar los datos que le pertenecen.

CryptDB está formado por dos elementos: el sistema de bases de datos y el servidor de aplicaciones; y tiene dos objetivos fundamentales: limitar el acceso a la información por parte del proveedor de la base de datos y mantener la confidencialidad en caso de que la seguridad de todo el sistema se viera comprometida.

Al igual que en la propuesta de Rodríguez, los campos son cifrados de distintas maneras en función del tipo de dato y la operación a realizar. Para ello, se define una estructura de capas: igualdad, orden, búsqueda y suma. Cada dato es cifrado tantas veces como sea necesario (mediante el algoritmo adecuado) en función de las operaciones que se puedan realizar sobre el mismo.

Los cifrados que se pueden encontrar en las capas son: aleatorio, determinístico, preservando el orden, homomórfico y para relaciones.

La arquitectura del sistema consta de dos partes: un proxy de base de datos y el sistema de gestión de bases de datos relacional sin modificar. El proxy se encarga de traducir las consultas al formato adecuado para su ejecución en el sistema de gestión de bases de datos cifrado. En dicho servidor, se almacena la estructura de las tablas para realizar el proceso de traducción consultas. El servidor de base de datos tiene una serie de funciones definidas por el usuario para llevar a cabo algunas de las rutinas de cifrado/descifrado.

Al ser un sistema funcional, los creadores del mismo lo han podido probar como motor de base de datos para aplicaciones habituales en internet, como PHPbb o HOTCPR. Los resultados obtenidos muestran niveles de rendimiento y ejecución de consultas muy aceptables. A pesar de ello, hay un conjunto de consultas básicas, relacionadas con ordenaciones y uniones, que no pueden ejecutarse.

Por otra parte el tamaño de la base de datos aumenta significativamente, debido a que un mismo campo necesita ser cifrado en varias capas.

III. Análisis y enfoque NoSQL

Se puede identificar algunos problemas en las propuestas anteriores.

En el caso de los sistemas con cifrado en descanso (*at rest*, sección II-A), es necesario confiar en el proveedor puesto que es éste el que elige y custodia las claves de cifrado. Además, al ser sistemas de gestión de bases de datos puramente relacionales, no han sido diseñados para ser escalables bajo el modelo de computación en la nube; es sobre el cliente donde recae la responsabilidad de la escalabilidad, diseñando y adaptando el esquema de su base de datos de forma especial para poder escalar de forma limitada mediante particionamiento (*sharding*) u otras técnicas similares. No parece ser la mejor solución para el modelo *DaaS*.

Si bien la propuesta de Rodríguez (sección II-C) soluciona el problema de la confianza, puesto que es el cliente quien elige y custodia las claves de cifrado, presenta el mismo inconveniente frente a la escalabilidad horizontal del sistema. Por otra parte, el uso de criptografía homomórfica implica una gran carga computacional, si bien, en esta propuesta no se cuantifica suficientemente el impacto sobre el rendimiento del sistema no cifrado.

El sistema *CryptDB* (sección II-D) delega toda la gestión de seguridad al proxy traductor de consultas; por lo tanto, dicho proxy debe estar gestionado y alojado por el cliente para evitar tener que confiar en el proveedor *DaaS*. Bajo este modelo, el *back-end* relacional cifrado permite escalabilidad horizontal utilizando técnicas tradicionales como particionamiento (al igual que los sistemas con cifrado en descanso y la propuesta de Rodríguez) pero el proxy se convierte en un cuello de botella por el que pasan absolutamente todas las transacciones al *back-end* cifrado. Por otra parte, una gestión eficiente del proxy obligaría al cliente a tener y mantener su propio sistema de computación en cloud privado, que es, precisamente, lo que se quiere evitar al adoptar una estrategia *DaaS*.

Por todo lo anterior, consideramos que una posible solución a los sistemas *DaaS* seguros se encuentra en el uso de sistemas no exclusivamente relacionales (*NoSQL*, véase [1]), ya que están diseñados desde su origen para la escalabilidad horizontal en sistemas de computación en la nube.

Si bien existen funcionalidades equivalentes en ambos entornos, es necesario tener en cuenta que hay una serie de diferencias (véase [8]) entre ambos sistemas de bases de datos. En primer lugar, los datos no se almacenan en tablas sino en estructuras de documentos; no existen esquemas de tablas definidos en *NoSQL* ya que las estructuras pueden crecer de forma dinámica. Además, no existe un lenguaje estructurado de consultas estándar al estilo de *SQL* en los sistemas relacionales; podemos encontrar sistemas de bases

de datos *NoSQL* que tienen un lenguaje *UnQL* (Unstructured Query Language), pero la sintaxis difiere en cada una de ellas.

Los sistemas de bases de datos *NoSQL* están orientadas a cumplir el Teorema de Brewer o Teorema *CAP* (Consistency, Availability and Partition tolerance; véase [7]) y no el modelo *ACID* (Atomicity, Consistency, Isolation and Durability) propio de los sistemas relacionales transaccionales. No obstante, existen alternativas para solventar este inconveniente.

Al igual que en las propuestas de Rodríguez y *CryptDB*, el objetivo principal bajo el enfoque *NoSQL* consiste en que la base de datos pueda ser gestionada por un tercero manteniendo la confidencialidad de la información y permitiendo la realización de consultas sobre los datos como si estuvieran en claro. Para ello, resulta imprescindible el almacenamiento múltiple de cada campo con distintos tipos de cifrado en función de las operaciones a realizar sobre el mismo.

A continuación, detallamos algunas de las operaciones a considerar junto con sus posibles soluciones:

- *Alta, baja y modificación de documentos.* Es necesario que el cifrado se realice en el cliente, determinando los esquemas de cifrado oportunos en función del tipo de cada campo: cifrado homomórfico y cifrado que preserva el orden en el caso de campos numéricos y cifrado determinístico en caso de campos alfanuméricos.
- *Búsqueda de elementos.* Para el filtrado y localización de elementos se ha de considerar la naturaleza del campo sobre el que se está buscando para determinar que cifrado utilizar. Esta operación se realiza en el cliente.
- *Orden de elementos.* El uso de un cifrado que preserva el orden permite ordenar resultados y delimitar intervalos.
- *Suma de elementos.* Empleando un cifrador homomórfico para la suma, el servidor es capaz de realizar las operaciones de suma sin tener que descifrar los datos.
- *Relación entre documentos.* Consiste en la búsqueda de un elemento relacionado en dos colecciones de datos.

Por otra parte, es necesario encontrar una implementación adecuada para el cifrado de datos numéricos en coma flotante y de tipo fecha, de modo que se pueda operar con ellos en el servidor sin necesidad de descifrar los mismos.

IV. Conclusión

Se han descrito y analizado algunas propuestas significativas de sistemas de bases de datos seguros para la computación en la nube. También se ha introducido una posible solución a las deficiencias de los sistemas existentes mediante un enfoque no exclusivamente *SQL* (*NoSQL*).

Los sistemas *NoSQL*, al contrario que los sistemas de gestión de bases de datos relacionales tradicionales, han sido diseñados desde el origen para la computación en la nube; por lo tanto, posibilitan niveles de rendimiento y escalabilidad óptimos en dichas plataformas. No obstante, presentan ciertas dificultades como el hecho de almacenar la información en documentos sin estructura relacional o no ofrecer un lenguaje de consulta estándar como el *SQL*.

Por ello, este enfoque resulta una vía de trabajo futuro muy interesante que se está explorando en la actualidad.

Agradecimientos

Investigación parcialmente financiada por el MINECO mediante el proyecto TIN2011-25452.

Referencias

[1] L. Adam, J. Mattson, "Investigating storage solutions for large data: A comparison of well performing and scalable data storage solutions for real time extraction and batch insertion of data," master of science thesis, Department of Computer Science and Engineering, Chalmers University of Technology, Göteborg, Suecia, 2010.

[2] A. Alcalde, "Lo último en criptografía: Fully Homomorphic Encryption," 2013. Disponible en http://elbauldelprogramador.com/lo-ultimo-en-criptografia-fully-homomorphic-encryption

[3] A. Boldyreva, N. Chenette, Y. Lee, A. O'Neill, "Order-preserving symmetric encryption" en *Advances in Cryptology- EUROCRYPT 2009*, Praga, República Checa, LNCS vol.5479, 2009, pp. 224-241.

[4] C. Curino, E.-P.-C. Jones, R.-A. Popa, N. Malviya, E. Wu, S. Madden, H. Balakrishnan, N. Zeldovich, "Relational cloud: A database-as-a-service for the cloud," en *Proceedings of the 5th Biennial Conference on Innovative Data Systems Research*, Pacific Grove, CA, 2011, pp. 235–241.

[5] K. Van Gelder, "Elastic Data Warehousing in the Cloud," Report Faculty of Exact Sciences, Vrije Universiteit Amsterdam, Netherlands, 2011. Disponible en http://homepages.cwi.nl/~boncz/msc/2011-KeesvanGelder.pdf

[6] C. Gentry, "A fully homomorphic encryption scheme", tesis doctoral, Departament of Computer Science, Stanford University, 2009.

[7] S. Gilbert, N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," SIGACT News, vol. 33 Iss. 2, pp. 51–59, 2002.

[8] L. P. Issac, "SQL vs NoSQL Database Differences Explained with few Example DB," The Geek Stuff, 2014. Disponible en http://www.thegeekstuff.com/2014/01/sql-vs-nosql-db/

[9] Microsoft docs., "Cifrado de datos transparente (TDE),". Disponible en http://technet.microsoft.com/es-es/library/bb934049.aspx

[10] Oracle docs., "Transparent Data Encryption," Oracle Database Advanced Security Administrator's Guide. Disponible en http://docs.oracle.com/cd/B19306_01/network.102/b14268/asotrans.htm

[11] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," en *Advances in Cryptology- EUROCRYPT 2009*, Praga, República Checa, LNCS vol.1592, 2009, pp. 223-238.

[12] R. A. Popa, N. Zeldovich, H. Balakrishnan, "CryptDB: A Practical Encrypted Relational DBMS," Report MIT-CSAIL-TR-2011-005, Computer Science and Artificial Intelligence Laboratory, Cambridge, 2011.

[13] R. A. Popa, C. Redfield, N. Zeldovich, H. Balakrishnan, "Cryptdb: protecting confidentiality with encrypted query processing," en *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, Cascais, Portugal, 2011.

[14] R. A. Popa, N. Zeldovich, "Cryptographic treatment of CryptDB's Adjustable Join," Report MIT-CSAIL-TR-2012-006, Computer Science and Artificial Intelligence Laboratory, Cambridge, 2012.

[15] R. A. Popa, F. H. Li, N. Zeldovich, "An ideal-security protocol for order-preserving encoding," en *Proceedings of 2013 IEEE Symposium on Security and Privacy*, 2013, pp. 463–477.

[16] L.M.X. Rodríguez, "Esquema de cifrado para la ejecución de consultas en bases de datos cifradas," tesis doctoral, Departamento de Computación del Centro de Investigación y Estudios Avanzados del Instituto Politécnico Nacional, México, D.F., Diciembre, 2009.

[17] S. Tu, M. F. Kaashoek, S. Madden, N. Zeldovich, "Processing analytical queries over encrypted data," en *Proceedings of the 39th international conference on Very Large Data Bases*, Riva del Garda, Italia, 2013, pp. 289–300.

Bitcoins y el problema de los generales bizantinos

Cristina Pérez-Solà
Departament d'enginyeria de la informació i les comunicacions
Universitat Autònoma de Barcelona
Email: cperez@deic.uab.cat

Jordi Herrera-Joancomartí
Departament d'enginyeria de la informació i les comunicacions
Universitat Autònoma de Barcelona
Email: jordi.herrera@uab.cat

***Resumen*—En este artículo pretendemos mostrar porqué, a nuestro entender, la comunidad científica y en especial los que trabajamos en el ámbito de la criptografía y la seguridad de la información, debemos comprender el funcionamiento de la moneda digital Bitcoin. Como se verá, los motivos que presentamos trascienden a la propia moneda Bitcoin y se centran en la red peer-to-peer (P2P) subyacente a dicha moneda, que proporciona un sistema distribuido que permite mantener un registro público también distribuido. Dicho registro permite distintos usos y, como se verá, deja la puerta abierta a múltiples innovaciones.**

***Palabras clave*—Bitcoin, criptomoneda (*cryptocurrency*), P2P, Sistemas Distribuidos (*Distributed Systems*), Problema de los Generales Bizantinos (*Byzantine Generals Problem*)**

I. Introducción

Los sistemas distribuidos presentan un sinfín de propiedades que los hacen unos candidatos idóneos en distintos escenarios. Por ejemplo, son sistemas altamente escalables, que pueden ofrecer rendimientos muy elevados. Por otro lado, en cuanto a la seguridad se refiere, un sistema distribuido presenta la ventaja de eliminar el único punto crítico que supone un sistema centralizado, así como la supremacía que implica el control de dicho punto crítico.

Sin embargo, uno de los problemas también de seguridad asociado a los sistemas distribuidos es la naturaleza poco *controlable* de las entidades que participan en el sistema distribuido. Las entidades que lo forman tienen cierto grado de autonomía y, por lo tanto, su comportamiento puede ser alterado, ya sea a causa de fallos no deseados dentro de la propia entidad, como a causa de la existencia de entidades con intereses contrarios al resto del sistema. Uno de los problemas de seguridad asociados a los sistemas distribuidos es el conocido como los generales bizantinos.

El **problema de los generales bizantinos** [1] es un experimento mental creado para ilustrar el dilema de lograr un consenso entre un conjunto de entidades con un objetivo común cuando entre ellas pueden existir traidores, es decir, entidades con objetivos opuestos que intenten dinamitar el proceso. Además, se supone que las comunicaciones entre dichas entidades son limitadas e inseguras. El problema se presenta como una analogía con un escenario de guerra, donde un grupo de generales bizantinos se encuentran acampados con sus tropas alrededor de una ciudad enemiga que desean atacar. Después de observar el comportamiento del enemigo, los generales deben comunicar sus observaciones y ponerse de acuerdo en un plan de batalla común que permita atacar la ciudad y vencer. Para ello, los generales se comunican únicamente a través de mensajeros. Además, existe la posibilidad que algunos de los generales sean traidores y, por lo tanto, decidan enviar mensajes con información errónea con el objetivo de confundir a los generales leales. Un algoritmo que solucione el problema debe asegurar que todos los generales leales acuerdan un mismo plan de acción y que unos pocos traidores no pueden conseguir que el plan adoptado por los generales leales sea equivocado.

Uno de los grandes logros que supone Bitcoin, más allá de ser la primera criptomoneda con una aceptación extendida[1] por todo el mundo, es el hecho de ofrecer la primera solución práctica al problema de los generales bizantinos. La aplicación de los generales bizantinos a la criptomoneda permite, por primera vez en la historia, transferir propiedad digital a otro usuario de Internet, de manera que solo el propietario pueda hacerlo, únicamente el destinatario pueda recibirla, todo el mundo pueda validar la transferencia y esta sea reconocida por todos los participantes, todo ello realizado de manera totalmente distribuida.

En este artículo, expondremos porqué es interesante conocer la criptomoneda Bitcoin y repasaremos las aportaciones que el esquema utilizado por Bitcoin representan, más allá de la propia moneda.

El resto del artículo se estructura de la siguiente manera: la Sección II presenta a grandes rasgos el sistema Bitcoin; después, la Sección III enfatiza las características de Bitcoin en relación a la notarización de información; posteriormente, la Sección IV comenta extensiones de la notarización que se han propuesto, tanto como para el propio sistema Bitcoin como para sistemas posteriores construidos a su imagen; seguidamente, la Sección V menciona algunas de las aplicaciones que un sistema de notarización distribuido puede tener; finalmente, la Sección VI presenta las conclusiones.

II. Bitcoin: conceptos básicos

Dado que este artículo pretende resaltar las características que hacen del sistema Bitcoin un sistema a tener en cuenta

[1] Trabajos existentes realizados con datos de Enero de 2014 [2] descubren alrededor de 110000 nodos diferentes conectados en un día cualquiera.

en distintos ámbitos más allá de la propia moneda, en esta sección se describen únicamente unas nociones muy básicas del funcionamiento de los Bitcoins, imprescindibles para que el lector comprenda el abasto de las contribuciones que Bitcoin representa.[2] Por este motivo, es posible que dicha descripción sea incluso insuficiente para entender la corrección y completitud del sistema Bitcoin como moneda digital. El lector interesado en conocer a fondo el funcionamiento de la moneda puede obtener más información en: [3], [4], [5].

II-A. Las transacciones

La unidad básica de funcionamiento de Bitcoin son las llamadas **transacciones**. Una transacción indica un movimiento de Bitcoins de una dirección de origen a una dirección de destino. Cada **dirección** de Bitcoins representa una clave pública (Bitcoin se basa en criptografía de curvas elípticas). Para **gastar** Bitcoins es necesario conocer la clave privada asociada a la clave pública que contenga un saldo en Bitcoins. Entonces, se pueden gastar esos Bitcoins, es decir, transferirlos a otra dirección, firmando digitalmente con la clave privada la transmisión de esta información y enviando la nueva transacción a toda la red. Veámoslo con un ejemplo:

Sea $\{PK_A, SK_A\}$ ($\{PK_B, SK_B\}$) el par de claves, pública y privada, del usuario Alice (respectivamente, del usuario Bob). La función $Addr(PK)$ nos devuelve la dirección de Bitcoin asociada a la clave pública PK, H es una función hash y $Sig_{SK}(m)$ representa la firma de m con la clave privada SK. Supongamos que Alice ha recibido anteriormente en una transacción T_0 la cantidad de $25BTC$ a su dirección, $Addr(PK_A)$:

$$\begin{aligned} T_0 &= \{input_0, output_0\} \\ input_0 &= \{\cdots\} \\ output_0 &= \{Addr(PK_A), 25\} \end{aligned}$$

Alice desea, entonces, enviar los $25BTC$ a Bob. Para ello, Alice crea una nueva transacción, T_1:

$$\begin{aligned} T_1 &= \{input_1, output_1\} \\ input_1 &= \{H(T_0), Sig_{SK_A}(T_0 + output_1), PK_A\} \\ output_1 &= \{Addr(PK_B), 25\} \end{aligned}$$

Veamos el motivo de incluir cada uno de los elementos en la transacción. En primer lugar, la transacción nueva T_1 incluye el hash de la transacción que se quiere gastar, T_0, que actúa como un puntero. En segundo lugar, Alice, que es la propietaria de la dirección que contiene los fondos, es la única que puede gastarlos ya que es la única que conoce la clave privada SK_A necesaria para realizar la firma $Sig_{SK_A}(T_0 + output_1)$. Además, si Alice no ha usado anteriormente esta dirección, ella es también la única que conoce su clave pública PK_A, ya que la función $Addr$ es pública pero no invertible. Por este motivo, para que se pueda validar la firma, la transacción debe incluir PK_A. Por último, Alice indica que quiere transferir los fondos a Bob firmando la dirección de Bob juntamente con el importe a transferir ($output_1$). De este modo, solamente Bob, que es el único conocedor de su clave privada, podrá gastar la transacción T_1.

Bob puede verificar que le han sido transferidos los fondos comprobando que $Addr(PK_A)$ coincida con la dirección de destino de T_0 y que la firma $Sig_{SK_A}(T_0{+}output_1)$ es correcta con PK_A.

II-B. La cadena de bloques

Tal como hemos descrito el sistema hasta este punto, no hay nada que impida a Alice gastar repetidamente los $25BTC$ que ha recibido en la transacción T_0, es decir, crear $T_1, \ldots, T_i$ transacciones con direcciones de destino diferentes utilizando la misma dirección de origen y el mismo puntero a la transacción anterior. Este comportamiento se conoce bajo el nombre de **doble gasto** y, obviamente, es necesario prevenirlo en cualquier tipo de moneda virtual.

Con el objetivo de prevenir el doble gasto, Bitcoin anota todas las transacciones ocurridas en un registro común conocido como **cadena de bloques** (o *blockchain*). De este modo, cuando Bob recibe la transacción T_1 de Alice, puede acudir al registro público y comprobar que Alice no haya gastado anteriormente el dinero que le está transfiriendo, es decir, comprobar que no existe ninguna otra transacción que tiene en su $input$ el mismo valor $H(T_0)$.

Este registro único se genera, distribuye y almacena de forma distribuida, de modo que todos los participantes están de acuerdo en su contenido sin la intervención de ninguna autoridad central. Es en esta creación de un registro público único de manera distribuida donde Bitcoin resuelve de manera práctica el problema de los generales bizantinos y por el cual el potencial de Bitcoin sobrepasa de largo el de una moneda virtual.

El registro público de Bitcoin (la cadena de bloques) está formado, como su nombre indica, por un conjunto de **bloques** enlazados de manera secuencial. Con el paso del tiempo, nuevos bloques son creados y añadidos a la cadena existente. La cadena de bloques es, por lo tanto, un registro que solo permite anexar información. Cada bloque contiene una cabecera y una carga útil. La carga útil son las transacciones que han ocurrido en el sistema desde que se creó el último bloque. De este modo, el conjunto de transacciones aceptadas como válidas por la red son las transacciones contenidas en cada uno de los bloques que pertenecen a la cadena de bloques. A su vez, la cabecera de cada bloque contiene un puntero al bloque anterior, de modo que los bloques forman una cadena. Además, la cabecera contiene también un valor de *nonce*, que permite crear bloques válidos como veremos a continuación.

Los usuarios que se dedican a crear bloques en la red Bitcoin son conocidos como **mineros**, y son una pieza fundamental del esquema. Cualquier usuario de la red puede ser un minero. Su trabajo consiste en validar las transacciones que se envían por la red P2P, incluyendo las válidas en nuevos bloques y

[2] De hecho, se presenta una simplificación del esquema que no corresponde exactamente al protocolo Bitcoin, pero que permite entender sus puntos clave sin entrar en todos los detalles.

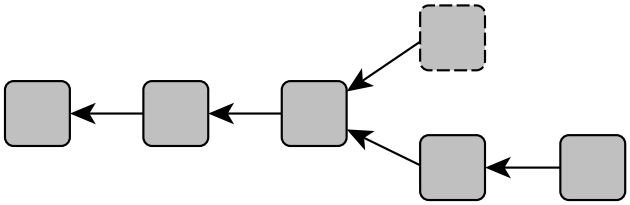


Figura 1. Bifurcación de la cadena.

descartando las inválidas. De este modo, si una transacción intenta gastar un importe ya gastado, o bien un usuario intenta gastar una transacción que no le pertenece (generando por lo tanto una firma inválida), esta nueva transacción nunca será incluida en un bloque y, de este modo, no habrá existido para el sistema. Por lo tanto, se necesita asegurar que los mineros hacen su trabajo correctamente, es decir, que aunque existan algunos mineros *traidores* que actúen en contra del interés común, se asegura que los mineros *leales* consigan acordar una cadena única, que contenga únicamente transacciones válidas.

Para lograrlo, se requiere que los bloques contengan una **prueba de trabajo** (*proof-of-work*) para ser considerados válidos. Dicha prueba de trabajo demuestra que el minero ha gastado un tiempo de computación en la generación del bloque. De este modo, mientras el poder de cómputo de la red esté distribuido, un grupo pequeño de mineros *traidores* no podrá modificar la cadena de bloques en su beneficio. La prueba de trabajo que utiliza Bitcoin consiste en encontrar un valor de *nonce* para el nuevo bloque de tal manera que el hash del bloque sea inferior a un valor objetivo fijado. Por las propiedades de las funciones hash, la única manera de conseguir un hash inferior al valor objetivo es ir probando diferentes valores de *nonce*, hasta dar con uno que genere el hash buscado.

Una vez un minero ha encontrado un bloque que cumple los requisitos, lo envía a toda la red, de manera que el nuevo bloque se convierte en el último de la cadena. A partir de ese momento, todos los mineros actualizan su estado, es decir, actualizan el puntero al último bloque conocido y actualizan las transacciones conocidas por el sistema, incluyendo en la generación del nuevo bloque solo aquellas que no se encuentran ya en la cadena.

Se puede dar el caso que dos mineros encuentren dos bloques distintos válidos que tengan el mismo bloque padre de manera más o menos simultánea (Figura 1), y que ambos envíen los bloques encontrados a toda la red. En este caso, se acepta el bloque que genere la cadena más larga, en términos del trabajo invertido en realizarla.

Como se ha visto, la existencia de mineros es fundamental para el funcionamiento del sistema, así que es necesario asegurar que existen **incentivos** suficientes para que los usuarios de la red quieran realizar el trabajo de minería, cosa que supone un coste (al menos en electricidad) para ellos. Actualmente el incentivo principal de los mineros es la recompensa que reciben, en forma de Bitcoins, cada vez que generan un bloque. Hemos visto como se transferían Bitcoins de una dirección a otra pero, hasta este momento, no hemos comentado como se crean estos Bitcoins. Los Bitcoins se crean a partir de un tipo de transacción especial, la **transacción de generación**, que se incluye en cada bloque. Dicha transacción tiene una dirección de destino (que pertenece al minero que se ha generado el bloque) pero no tiene ninguna dirección de origen. El importe de esta transacción de generación va disminuyendo con el tiempo y, a día de hoy, es de $25BTC$[3]. Cada bloque solo puede contener una única transacción de generación.

III. Notarización de información en el protocolo Bitcoin

En esta sección, describiremos las contribuciones de Bitcoin con relación a la notarización de información, es decir, a la creación de un registro único común de manera totalmente distribuida.

Bitcoin utiliza la cadena de bloques para almacenar transacciones, es decir, las unidades de información almacenadas en el registro único de Bitcoin son transacciones. Estas transacciones han sido creadas con anterioridad por algún miembro de la red, y difundidas por toda la red.

Suponiendo que existen usuarios en la red creando transacciones, el trabajo de los nodos de Bitcoin, es decir, de las entidades que forman parte del protocolo distribuido para crear el registro común de información, se resume en cuatro grandes tareas: validación, afianzamiento, transmisión y almacenaje.

III-A. Validación

Los mineros validan cada una de las transacciones que se incluyen en un bloque. Sea T_1 la transacción a validar, las comprobaciones a realizar son las siguientes:

- No existe doble gasto, es decir, T_1 no intenta gastar una transacción anterior T_0 ya gastada anteriormente.
- La transacción anterior T_0 que se intenta gastar existe.
- La clave pública especificada en la entrada de T_1 se corresponde a la dirección de salida especificada en T_0.
- La firma es correcta al validarla con la clave pública especificada en la entrada de T_1.

Aunque el funcionamiento de Bitcoin es muy similar al que hemos descrito, en realidad su especificación no se describe en estos términos sino en otros mucho más generales, con el objetivo de permitir realizar transacciones más complejas. En vez de fijar como se deben codificar las claves públicas, las direcciones y las firmas dentro de cada transacción, Bitcoin dispone de un **lenguaje de scripting** propio basado en pila, el código del cual se inserta tanto en las salidas como en las entradas de las transacciones. A la hora de validar una transacción, se apila el *script* de entrada con el de salida y se evalúa el *script* resultante. Si el resultado final de la evaluación es Cierto, entonces la transacción se considera válida. En caso contrario, la transacción se considera inválida.

[3]En el momento de escribir estas líneas, en Febrero de 2014, este importe equivale a unos 20,000 dólares.

Las validaciones descritas en este apartado forman parte de lo que sería una validación completa. Nótese que para realizar esta validación es necesario conocer la cadena de bloques entera, juntamente con todas las transacciones que contiene. Esto tiene un coste de espacio elevado. Además, recorrer la cadena en busca de las transacciones implicadas supone también un coste computacional elevado, que junto al coste en espacio, suponen un problema para dispositivos ligeros como móviles o incluso ordenadores limitados. Por este motivo, Bitcoin dispone del **Protocolo de Validación Simplificado** (SPV, del inglés *Simplified Payment Verification*), que permite a un usuario comprobar que ha recibido un pago utilizando significativamente menos recursos a costa de una reducción en la seguridad de la validación. Para la validación SPV, solo es necesario disponer de una copia de las cabeceras de los bloques de la cadena (que el cliente puede pedir a otro(s) nodo(s) de la red en cualquier momento) así como de algunos valores hash que permiten localizar la transacción dentro del bloque.

III-B. Afianzamiento

Además de validar las transacciones, es necesario también crear los bloques que las afianzan, así como validar a su vez la corrección de estos bloques. Este proceso es el que permite construir el registro común único y se realiza de manera totalmente distribuida en Bitcoin.

Como hemos visto, el afianzamiento en Bitcoin se basa en una prueba de trabajo (*proof-of-work*), consistente en encontrar un valor de *nonce* para el bloque B de tal manera que $H(B) < t$, es decir, que el hash del bloque sea inferior al objetivo fijado. El valor objetivo no es constante, y permite adaptar la dificultad de la prueba al poder de cómputo de la red en cada momento, con el propósito de generar un nuevo bloque cada 10 minutos.

De este modo, si un atacante quiere modificar la cadena de bloques, ya sea para dar marcha atrás y anular transacciones que ha realizado, ya sea para tener control de lo que se anota en el registro común, será necesario que éste disponga de un poder de cómputo superior al $50\,\%$ de la red[4]. En caso contrario, si el atacante intenta modificar la cadena de bloques generando sus propias alternativas, no tendrá suficiente poder de cómputo como para generar bloques más rápido que el resto de la red, por lo que su rama no será la más larga, y será descartada.

III-C. Transmisión

Bitcoin utiliza una red P2P totalmente distribuida para propagar la información. Bloques y transacciones son transmitidos a través de esta red.

Cuando un nodo quiere realizar una transacción (o bien encuentra un bloque válido), este lo envía a toda la red. Para hacerlo, lo envía a los nodos que se encuentran directamente conectados con él y éstos, a su vez, lo reenvían a sus vecinos, siempre que el objeto en cuestión (bloque o transacción) sea válido. De este modo, la información se propaga por toda la red.

Dado que, a diferencia de los bloques, las transacciones no contienen ninguna prueba de trabajo, un nodo malicioso podría crear un gran número de transacciones válidas con la intención de desbordar la red. Para evitar este tipo de ataques, los nodos estándar de Bitcoin aplican una política de retransmisión de transacciones, que obliga a incorporar una **comisión** a las transacciones que cumplen ciertas características que las hacen ideales para este tipo de ataques. Aún así, los usuarios que realizan transacciones tienen libertad para decidir si pagan o no una comisión y, en caso de hacerlo, del importe que esto conlleva. Estas comisiones afectan, como hemos comentado, la retransmisión de la transacción, además de su inclusión en un bloque. Esto último es debido a que el minero, además de cobrar la recompensa por encontrar un bloque, también obtiene todas las comisiones que las transacciones que contiene el bloque incorporan. Por este motivo, incluir comisiones en las transacciones puede crear incentivos adicionales para que los mineros las incluyan en sus bloques.

III-D. Almacenaje

El almacenaje de la cadena de bloques se lleva a cabo con mucha redundancia: todos los nodos completos de la red contienen una copia entera de la cadena de bloques (y sus transacciones). Esto permite a estos nodos validar de manera correcta cada nueva transacción.

Tener que mantener una copia completa de la cadena puede suponer un problema para los nodos operando en dispositivos ligeros como, por ejemplo, dispositivos móviles. En Febrero de 2014, después de 5 años de operación de la moneda Bitcoin, la cadena de bloques ocupa unos 13 GB.

IV. Extensiones para la notarización de la información

En esta sección, repasaremos algunas de las mejoras o alternativas que se han propuesto sobre el protocolo de Bitcoin, algunas de ellas implementadas ya en otras criptomonedas, otras solo presentadas a nivel teórico.

IV-A. Validación

Aunque los *scripts* de Bitcoin permiten especificar qué se necesita para poder gastar una transacción, el lenguaje es limitado. Según la propia descripción del lenguaje, este **no** es **Turing-completo** por diseño, argumentando motivos de seguridad para justificar esta decisión. Si bien es cierto que esto previene de realizar ciertos ataques (pensemos, por ejemplo, en un *script* con un bucle infinito, que se ejecutaría de manera indefinida cada vez que se intentara validar), también limita el conjunto de programas que se pueden codificar con él.

Una extensión que se ha propuesto en este sentido es incorporar un lenguaje Turing-completo a las transacciones [7], aumentando así la potencia de las mismas. Este lenguaje debe ir acompañado de un sistema de seguridad que permita evitar ciertos ataques, como el anteriormente comentado *script* de ejecución infinita. Una de las propuestas contempla incluir

[4]Estudios recientes presentan un ataque teórico que reduce este valor al $33\,\%$ del poder de cómputo total[6].

una comisión que se debe pagar por cada paso de ejecución del algoritmo, de manera que los *scripts* más simples, que suponen menos tiempo de validación, resulten más baratos que aquellos más complejos, que necesitan gastar tiempo de computación para ejecutarse.

Otra de las limitaciones del protocolo Bitcoin se encuentra en relación al Protocolo de Validación Simplificado. El protocolo se puede llevar a cabo para el tipo de transacción estándar dentro de Bitcoin, pero se complica enormemente (hasta el punto que no se ha encontrado solución aún) para ciertas variaciones del esquema.

IV-B. Afianzamiento

Bitcoin utiliza una prueba de trabajo basada en el cálculo de hashes para afianzar la información. A día de hoy[5], se estima que la red dispone de un poder de cómputo superior a los 23000 TH/s. Esto supone un gasto energético elevado, hecho que ha empezado a causar alarma por los posibles efectos negativos sobre el medio ambiente. Además, dicho gasto energético únicamente se utiliza para la propia criptomoneda ya que el cálculo de los hash para afianzar los bloques no tiene ningún otro fin. Por lo tanto, es interesante plantearse alternativas a la prueba de trabajo basada en hash que permitan obtener una funcionalidad equivalente. Se han propuesto cuatro enfoques diferentes:

Proof-of-Work: Como hemos visto, consiste en demostrar que se ha realizado una cantidad de trabajo para conseguir el bloque. Por lo tanto, la probabilidad de conseguir minar un bloque depende del poder de cómputo empleado en el trabajo. En este ámbito, las mejoras se centran en dos alternativas. Por un lado, proponer funciones que no requieran una inversión en hardware para el minado de bloques (como sucede actualmente con la función SHA256), para democratizar el proceso de minado y evitar así grandes clústers de minado que pudieran llegar a controlar la red. Dentro de esta alternativa se encuentran funciones hash, como por ejemplo scrypt [8] que requieren un volumen elevado de memoria para su cálculo, haciendo poco viable la creación de hardware específico. Otro enfoque, mucho más ambicioso, es la propuesta de una función de *proof-of-work* tal que su propio cálculo permita resolver problemas útiles computacionalmente costosos. El problema principal de este enfoque es formalizar problemas que tengan las siguientes propiedades, necesarias para una *proof-of-work* utilizada como sistema de validación de los bloques de la cadena: 1) verificabilidad: el problema propuesto debe ser difícil de realizar pero, una vez resuelto, la validación de la solución encontrada debe ser muy simple; 2) granularidad: la dificultad del problema propuesto debe ser granular, en el sentido que se debe permitir ajustar la dificultad del mismo de forma controlada y progresiva. En la actualidad únicamente se conoce una *proof-of-work* con estas características, utilizada en la moneda digital PrimeCoin [9]. En este caso, la *proof-of-work* consiste en encontrar ciertas cadenas de números primos, en concreto, cadenas de Cunningham de primera y segunda especie o cadenas de primos gemelos.

Proof-of-Stake: En este caso, la probabilidad que un minero encuentre un bloque depende de la cantidad de Bitcoins que posee actualmente. De este modo, mientras la posesión de Bitcoins sea distribuida, también lo será la capacidad de minar.

Proof-of-Burn: En este tipo de pruebas, la probabilidad de conseguir afianzar un bloque depende del número de Bitcoins destruidos expresamente para este propósito, es decir, quemados. Destruir Bitcoins es tan sencillo como enviarlos a direcciones que no se puedan gastar, es decir, a *scripts* que se evalúen a Falso de manera deliberada.

Proof-of-Excellence: En este sistema definido vagamente en [10], se crean torneos periódicamente y se minan bloques en función del rendimiento de cada participante en el torneo.

IV-C. Transmisión

Algunas criptomonedas surgidas después del auge de Bitcoin modifican el tiempo medio necesario para crear un bloque, fijado en 10 minutos en Bitcoin. Aunque parezca un cambio trivial, esto tiene consecuencias importantes sobre la seguridad del esquema.

Por un lado, la seguridad de una transacción en Bitcoin se mide utilizando el número de **confirmaciones** que ésta tiene, es decir, cuántos bloques se han añadido a la cadena después del bloque que contiene la transacción en cuestión. El motivo es que, como más confirmaciones tenga una transacción, más difícil es anularla, ya que para ello habría que construir una rama alternativa de la cadena que supere en dificultad a la rama actual. Bajo este punto de vista, fijar un tiempo de creación de bloques de 10 minutos hace de Bitcoin un sistema lento en dar por válidas las transacciones. El cliente estándar espera a que existan 6 confirmaciones antes de aceptar una transacción como pago, lo que fijaría un tiempo medio de 1 hora para el proceso.

Por otro lado, cuando un nuevo bloque es encontrado por un minero, este lo envía a sus vecinos, de modo que el bloque se propaga por la red. Esta propagación no es instantánea, y son necesarios algunos segundos para que los nodos la reciban[2]. Durante este tiempo de propagación, el minero que ha encontrado el bloque ya se encuentra minando encima de este, mientras que el resto de mineros aún trabajan en el bloque anterior. Esto tiene dos consecuencias importantes. La primera es que estos últimos mineros están realizando trabajo inútil. El porcentaje de trabajo inútil por bloque, suponiendo un tiempo de propagación constante, es mayor como menor sea el tiempo de generación de bloques. La segunda consecuencia se deriva también de este problema, ya que el minero que ha encontrado el bloque se encuentra en clara ventaja respecto al resto de la red. Esto también se acentúa con la disminución del tiempo de generación de los bloques.

Una de las propuestas para minimizar el impacto que el trabajo inútil sobre bloques ya minados supone para el sistema es la de recompensar no solo al bloque que queda en la cadena principal, sino también a algunos de los bloques válidos que hayan quedado en otras bifurcaciones de la cadena [7].

[5]Febrero 2014

IV-D. Almacenaje

En relación al almacenaje de la información, el principal problema que Bitcoin tiene que afrontar es la escalabilidad. Con la continua creación de nuevas transacciones, el tamaño de la cadena de bloques no hace más que aumentar a buen ritmo, augurando problemas de almacenamiento a largo plazo. Por otro lado, las restricciones en el tamaño de los bloques implican que en la actualidad la red bitcoin solamente pueda procesar un máximo de 7 transacciones por segundo[6], un valor demasiado pequeño para una moneda con vocación global.

Aunque de momento no se ha implementado ninguna solución, se discute activamente la posibilidad de incorporar un algoritmo de poda de la cadena, de manera que no sea necesario guardar todas las transacciones. Así, transacciones antiguas podrían ser eliminadas, guardando de ellas solo su hash, para preservar la integridad de la cadena.

V. Posibles aplicaciones

Las utilidades prácticas de Bitcoin (o de un sistema basado en la cadena de bloques) sobrepasan de largo las de una simple moneda. A continuación, se listan algunas de las aplicaciones que el sistema proporciona, tanto aquellas de las que ya existen implementaciones sobre Bitcoin, como aquellas que surgen a partir de alternativas derivadas, así como también las que de momento quedan en un plano teórico.

- Submonedas ([7], [11], [10], [12]): La cadena de bloques se puede utilizar para representar transacciones de otros bienes, como por ejemplo, otras monedas, oro, acciones o propiedad.
- Derivados financieros ([7]): Se pueden representar también en la cadena de bloques derivados financieros, explicitando sobre que bien concreto se deriva el precio.
- Servicios de marca de tiempo o *timestamps* ([13], [11]): Incluyendo el hash de un archivo en un bloque de la cadena, se puede demostrar la existencia del archivo en el momento de la creación del bloque.
- Servicio de nombres de dominio o *DNS* ([11]): La cadena se puede utilizar también para almacenar información de nombres de dominio de manera totalmente distribuida.
- Sistemas de Reputación Anónimos ([7]): Del mismo modo que se pueden registrar nombres de dominio en la cadena, ésta se puede utilizar para construir sistemas de reputación anónimos.
- Cómputo multipartito seguro o *Secure multiparty computation* ([14], [15]): Protocolos para el cómputo bipartito y multipartito seguro se han propuesto recientemente, e incluso se han realizado implementaciones de algunos de los protocolos sobre Bitcoin.
- Juegos de Azar P2P ([16], [7]): Juegos de azar o loterías pueden implementarse de manera que éstos resulten seguros para todas las partes, utilizando trozos de la cadena (o hashes de estos) como generadores pseudoaleatorios.

[6]El tamaño máximo de un bloque es de 1MB y el tiempo entre bloques es de 10 minutos. Esto proporciona 1,7KB por segundo, lo que suponen unas 7 transacciones de 250bytes.

VI. Conclusión

Bitcoin es la primera moneda criptográfica que ha tenido una grande aceptación entre la población, existiendo implementaciones del cliente estándar que permiten operar con ella para múltiples plataformas. Este hecho, por si solo, ya tiene un gran mérito. Además, más allá de ser una criptomoneda en utilización, con un esquema criptográfico robusto, totalmente descentralizada y anónima, Bitcoin resuelve de manera práctica el problema de los generales bizantinos, permitiendo crear un registro único común de manera descentralizada. Los usos de este registro sobrepasan de largo los de la propia criptomoneda y, en consecuencia, creemos que es importante dar a conocer su existencia. Como hemos expuesto, ya existen diferentes iniciativas que hacen uso de este registro con finalidades muy diversas y, a nuestro parecer, estas iniciativas son solo el principio de una larga lista de aplicaciones que se pueden diseñar e implementar en base a este registro.

Agradecimientos

Este trabajo está parcialmente financiado por el Ministerio de Educación, a través de los proyectos TIN2011-27076-C03-02 CO-PRIVACY, TIN2010-15764 N-KHRONOUS, CONSOLIDER INGENIO 2010 CSD2007-0004 ARES, y de la beca FPU-AP2010-0078.

Referencias

[1] Lamport, Shostak, and Pease, "The Byzantine Generals Problem," in *Advances in Ultra-Dependable Distributed Systems, IEEE Computer Society Press*, 1995. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.1697

[2] J. A. D. Donet, C. Pérez-Solà, and J. Herrera-Joancomartí, "The Bitcoin P2P Network," in *Proceedings of the 1st Workshop on Bitcoin Research (in Assocation with FC14)*, ser. to appear, 2014.

[3] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," 2008. [Online]. Available: https://bitcoin.org/bitcoin.pdf

[4] M. Nielsen, "How the Bitcoin protocol actually works," 2013. [Online]. Available: http://www.michaelnielsen.org/ddi/how-the-bitcoin-protocol-actually-works/

[5] Bitcoin community, "Bitcoin wiki." [Online]. Available: https://en.bitcoin.it

[6] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable," in *Proceedings of the 1st Workshop on Bitcoin Research In Assocation with Financial Crypto*, 2014.

[7] V. Buterin, A. di Lorio, C. Hoskinson, and M. Alisie, "Ethereum white paper," 2013. [Online]. Available: http://www.ethereum.org

[8] C. Percival, "Stronger key derivation via sequential memory-hard functions," 2012.

[9] S. King, "Primecoin: Cryptocurrency with prime number proof-of-work," 2013. [Online]. Available: http://primecoin.io/bin/primecoin-paper.pdf

[10] S. King and S. Nadal, "Ppcoin," 2012. [Online]. Available: http://peercoin.net/bin/peercoin-paper.pdf

[11] vinced, "Namecoin: a secure general purpose p2p key/value storage system." [Online]. Available: http://namecoin.info/

[12] Bitcoin Wiki, "List of alternative cryptocurrencies," Última consulta: febrero 2014. [Online]. Available: https://en.bitcoin.it/wiki/List_of_alternative_cryptocurrencies

[13] Shesek, "BTProof: trusted timestamping on the bitcoin blockchain," 2013. [Online]. Available: https://www.btproof.com/

[14] A. M., D. S., M. D., and M. L., "Fair two-party computations via bitcoin deposits," in *Proceedings of the 1st Workshop on Bitcoin Research In Assocation with FC*, 2014.

[15] M. Andrychowicz, S. Dziembowski, D. Malinowski, and L. Mazurek, "Secure multiparty computations on bitcoin," in *to appear*, 2014.

[16] E. Voorhees, "Satoshidice," 2012. [Online]. Available: https://en.bitcoin.it/wiki/SatoshiDice

Evaluación del Rendimiento de una Solución de Cupones Electrónicos para Dispositivos Móviles

Andreu Pere Isern-Deyà, M. Francisca Hinarejos, Josep Lluís Ferrer-Gomila
Universitat de les Illes Balears (UIB), Email: {andreupere.isern, xisca.hinarejos, jlferrer}@uib.es

***Resumen*—El comercio electrónico móvil (m-commerce) representa ya una importante área de negocio con grandes oportunidades para consumidores y comerciantes. Sin embargo, todavía existen escenarios que requieren mejoras en cuanto a eficiencia, como son los cupones electrónicos. La eficiencia y el rendimiento de estas soluciones suele medirse únicamente considerando el coste de las operaciones criptográficas o realizando pruebas de laboratorio en entornos limitados, muchas veces una única máquina para ejecutar todo el escenario de pruebas (incluyendo consumidores y comerciantes). En este artículo presentamos un análisis del rendimiento de una solución de cupones electrónicos, mediante la cual comprobamos que no es suficiente analizar únicamente la carga debido a las operaciones criptográficas, sino que también deben considerarse otros factores, como el efecto de la red.**

***Index Terms*—cupón electrónico, seguridad, privacidad, eficiencia, análisis de rendimiento**

I. Introducción

El comercio electrónico (e-commerce) representa uno de los sectores más dinámicos e innovadores dentro de la economía global. Hoy día la atención que recibe el e-commerce es incluso mayor dado el auge de los dispositivos móviles y la mejora de las infraestructuras móviles de comunicaciones. De hecho, la cuota de mercado de usuarios con teléfonos inteligentes ha alcanzado alrededor del 18 % del total de dispositivos de usuario [1], siendo el comercio móvil (m-commerce) uno de los sectores más beneficiados por este hecho. De acuerdo con predicciones publicadas [2], en los próximos cuatro años las ventas *on-line* crecerán entre un 10 % y un 15 % anual.

Sin embargo, todavía hay mucho trabajo que hacer en el campo del m-commerce. Uno de los aspectos que más negativamente afecta a su crecimiento es la falta de privacidad y confianza de los consumidores respecto a los comerciantes y a las transacciones *on-line*. Otro de los aspectos importantes, a menudo dejado de lado, es la baja eficiencia (medida como el tiempo de respuesta) de las soluciones de m-commerce percibida por los consumidores.

En el campo del m-commerce, los cupones electrónicos son uno de los temas que requiere importantes mejoras, sobre todo en privacidad, usabilidad y eficiencia. Un cupón electrónico es la versión electrónica de los cupones en papel, documentos impresos que permiten al consumidor conseguir o acceder a productos o servicios, normalmente bajo un descuento o beneficio. En este sentido, encontramos conocidas soluciones comerciales, como cupones para restaurantes [3], [4], hoteles [5], etc., aunque todas ellas se basan en la utilización final del papel para poder canjearlos en los comercios. Este modo de funcionamiento conlleva una pérdida en tiempo y recursos tanto para los comerciantes como para los consumidores y además frena su expansión.

Tanto los cupones electrónicos individuales como los multicupones (el equivalente a los talonarios de cupones) han atraído la atención en los últimos años de la comunidad científica [6]–[20]. No obstante, las soluciones no se validan teniendo en cuenta el tiempo total de respuesta percibido por los consumidores, a pesar que este es un aspecto crítico que debe ser considerado tanto en la fase de diseño como de implementación.

Contribución. En este artículo incidimos en la importancia de analizar todos los costes que influyen en el tiempo de respuesta de los protocolos. Para realizar este trabajo, hemos implementado en Java una solución propia de multicupones electrónicos para múltiples comerciantes [16], llamada $\mathcal{MC} - 2\mathcal{D}$ y hemos analizado su eficiencia y rendimiento. La eficiencia la medimos respecto a una propuesta previa similar de multicupones electrónicos, la cual fue verificada en un entorno limitado usando una sola computadora. Gracias a este análisis, demostramos que nuestra propuesta mejora ampliamente la solución previa. Finalmente, desplegando la implementación en un entorno de producción con dispositivos móviles Android, servidores remotos y comunicaciones reales, analizamos su rendimiento teniendo en cuenta los efectos de la red. Además de comprobar que $\mathcal{MC} - 2\mathcal{D}$ es viable en un entorno real, también demostramos que no solo la criptografía incide en aumentar el tiempo de respuesta percibido por los clientes del m-commerce, sino que otros costes pueden ser incluso mayores.

Organización. El artículo está organizado de la siguiente forma. En la Sección II presentamos un análisis sobre las propuestas previas. En la Sección III se resume la solución propuesta. La Sección IV se dedica a presentar una comparación de eficiencia basada en el número y tipo de operaciones criptográficas. Utilizando un escenario real, en la Sección V analizamos los diferentes factores que influyen en el rendimiento de la solución. Finalmente, cerramos el trabajo con las conclusiones y las líneas futuras en la Sección VI.

II. Trabajos Previos

Como se ha comentado en la introducción, existen propuestas tanto comerciales como de carácter científico para cupones electrónicos. Respecto a las soluciones comerciales [3]–[5],

éstas intentar ofrecer soluciones sencillas a bajo coste y normalmente basadas en papel. Es decir, evitan en la medida de lo posible llevar a cabo nuevas inversiones en implementaciones y reutilizan los sistemas que ya tienen desarrollados. Por ejemplo, una manera sencilla que utilizan las compañías para ofrecer cupones electrónicos es desplegando una simple página web a través de la cual los clientes pueden comprar cupones para un determinado uso. Sin embargo, los clientes deben desplazarse físicamente hasta la tienda del comerciante (o a un establecimiento autorizado) o utilizar su propia impresora para obtener una copia en papel. Este modo de funcionamiento no facilita el uso de los cupones y limita su difusión.

Respecto a las propuestas científicas actuales referentes a cupones y multicupones electrónicos [6]–[20], éstas intentan ofrecer el mayor número de funcionalidades sin considerar el coste que pueden generar al aplicarse en escenarios reales. No obstante, solo un número reducido de ellas [13], [16] proponen soluciones para un entorno donde los usuarios puedan gastar sus cupones en diferentes comerciantes sin tener que emitir un multicupón para cada uno de ellos (escenarios multi-comerciante). Por otra parte, en muchas de las propuestas no se encuentra suficiente información para analizar su viabilidad, tanto la referente a la implementación de las operaciones criptográficas involucradas como a la eficiencia resultante. En la mayoría de los trabajos que proporcionan medidas de eficiencia, utilizan escenarios de laboratorio, con pruebas limitadas y sin tener en cuenta todos los factores que influyen en el rendimiento final de las soluciones. Los autores de [8], [10] analizan sus propuestas basándose en cómo el número de cupones contenidos en un multicupón incrementa el coste de los diferentes protocolos para su gestión. Como resultado, los autores en [8] afirman que el coste es lineal respecto al número de cupones involucrados en cada transacción, mientras que en [10] se afirma (sin aportar ninguna prueba) que su esquema tiene un coste computacional constante con independencia del número de cupones involucrados en cada transacción. En [21], el autor realiza una implementación del esquema presentado en [13], pero considerando un entorno de pruebas local, en donde todas las operaciones se ejecutan sobre una misma computadora. Otras propuestas de cupones electrónicos [6], [15], [17]–[19] proporcionan resultados de eficiencia en entornos muy limitados y en pocos casos consideran el uso de algún tipo de dispositivo móvil [20].

Por lo tanto, la evaluación de soluciones para multicupones se ha realizado principalmente en términos del número de operaciones criptográficas. Sin embargo, este tipo de análisis no puede asegurar la viabilidad de una propuesta sobre redes y dispositivos reales. Este es un aspecto crítico para el éxito del m-commerce en general y de las soluciones para multicupones electrónicos en particular.

III. Una Solución de Cupones Electrónicos

A continuación resumimos los puntos fundamentales de $\mathcal{MC}-2\mathcal{D}$, solución de multicupones electrónicos para escenarios multi-comerciante. Los detalles de la misma, así como un amplio análisis de seguridad se pueden consultar en [16].

III-A. ¿Cómo Proporcionar Privacidad?

El funcionamiento de la solución se basa en el uso de la firma parcial ciega y la firma de grupo.

Firma parcial ciega. Es una generalización de la firma ciega [22] en la que el firmante tiene la capacidad de añadir a la firma resultante un conjunto de datos comunes acordados previamente entre el firmante y el solicitante. $\mathcal{MC}-2\mathcal{D}$ usa el esquema de firma parcial ciega presentado en [23].

Firma de grupo. Es una primitiva criptográfica que genera firmas en las que la identidad de un firmante que pertenece a un grupo de usuarios se mantiene en secreto. En estos esquemas se define una tercera parte, llamada *Gestor de grupo*, que es el encargado de generar los parámetros necesarios para realizar estas firmas. Además, es la única entidad capaz de revocar el anonimato y revelar la identidad de la entidad firmante. En nuestra solución se usa el esquema de firma de grupo propuesto en [24].

III-B. Arquitectura y Protocolos

La Figura 1 representa la arquitectura de la solución $\mathcal{MC}-2\mathcal{D}$. Los participantes involucrados son el cliente ($\mathcal{C}$), el vendedor ($\mathcal{V}$), el emisor ($\mathcal{E}$) y el gestor de grupo ($\mathcal{G}$). Entre cada uno de los participantes se definen siete protocolos: `Inicialización`, `Afiliación/Desafiliación`, `Registro`, `Emisión`, `Pago Múltiple`, `Depósito` y `Reembolso`.

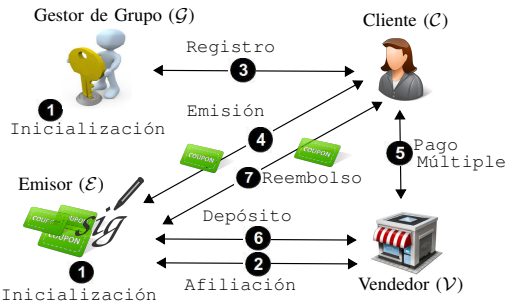


Figura 1: Arquitectura de $\mathcal{MC}-2\mathcal{D}$.

1. Inicialización. Tanto $\mathcal{G}$ como $\mathcal{E}$ inicializan sus servicios para recibir peticiones. $\mathcal{G}$ crea un conjunto de claves secretas y una clave pública para el esquema de firma de grupo, mientras que $\mathcal{E}$ y los clientes generan sus propias claves RSA.

2. Afiliación / Desafiliación. Todos los vendedores interesados en aceptar cupones emitidos por $\mathcal{E}$ se afilian a $\mathcal{E}$ mediante un simple acuerdo sin que se lleve a cabo ningún intercambio de información sensible.

3. Registro. Cada cliente interesado en usar cupones tiene que registrarse con $\mathcal{G}$ mediante el protocolo de `Registro` usando su identidad real, para así obtener una pareja de claves de grupo. Entonces $\mathcal{G}$ enlaza la identidad real de $\mathcal{C}$ con su correspondiente clave secreta para poder revocar el anonimato en caso de ser necesario.

4. Emisión. El protocolo permite a $\mathcal{C}$ solicitar a $\mathcal{E}$ la emisión de un multicupón firmado, al que llamamos $\mathbb{MC}^{2D}$.

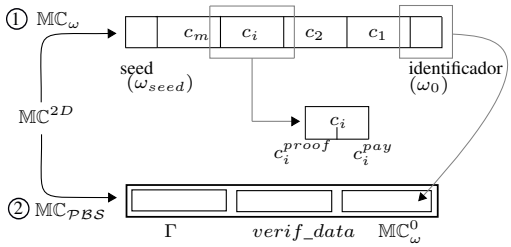


Figura 2: Estructura $\mathbb{MC}^{2D}$ compuesta por $\mathbb{MC}_\omega$ (considerando un sola tira de m cupones) y $\mathbb{MC}_{\mathcal{PBS}}$.

La estructura $\mathbb{MC}^{2D}$ (Figura 2) está compuesta por dos elementos principales: $\mathbb{MC}_\omega$ y $\mathbb{MC}_{\mathcal{PBS}}$.

1. $\mathbb{MC}_\omega$. Es la estructura que define todos los cupones que forman un multicupón. $\mathbb{MC}_\omega$ se organiza en múltiples tiras de cupones, cada una de ellas con un número determinado de cupones y con el mismo valor (o descuento). Para cada tira de m cupones, la solución genera iterativamente (mediante *hash chain*) $2m+1$ hashes desde un identificador aleatorio y secreto (*identificador seed*: ω_{seed}) hasta el último hash de la tira (*identificador de la tira*: ω_0). Entonces, cada cupón (c_i) se define mediante dos hashes: el de la derecha es la *información de pago* $(c_i^{pay} = \omega_{2i-1})$ y el de la izquierda es la *información de prueba* $\left(c_i^{proof} = \omega_{2i}\right)$, $\forall 0 < i \leq m$ (i indica el i-*ésimo* cupón de la tira). $\mathcal{C}$ mantiene $\mathbb{MC}_\omega$ en secreto, excepto el elemento ω_0, como veremos a continuación.
2. $\mathbb{MC}_{\mathcal{PBS}}$. Es la firma parcial ciega sobre $\mathbb{MC}_\omega^0$, la lista de todos los ω_0 contenidos en $\mathbb{MC}_\omega$. El elemento $\mathbb{MC}_{\mathcal{PBS}}$ contiene además datos de verificación $(verif\_data)$ así como información pública y acordada previamente (Γ) entre $\mathcal{C}$ y $\mathcal{E}$. Ésta define las características de $\mathbb{MC}^{2D}$: número de tiras y número de cupones en cada tira, el valor o descuento de cada cupón, marcas temporales para limitar su validez, etc.

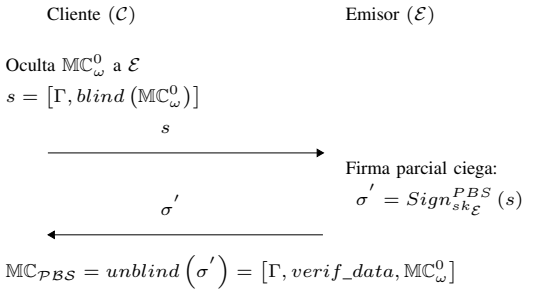


Figura 3: Protocolo de `Emisión`.

Una vez $\mathcal{C}$ ha generado $\mathbb{MC}_\omega$, empieza el protocolo de `Emisión` (Figura 3). La emisión de un multicupón implica la ejecución de un proceso basado en una firma parcial ciega sobre $\mathbb{MC}_\omega^0$, mediante el cual $\mathcal{E}$ firma $\mathbb{MC}_\omega^0$ (aunque no pueda obtener los datos en claro), juntamente con la información común (Γ). Como resultado, $\mathcal{C}$ obtiene $\mathbb{MC}_{\mathcal{PBS}}$, elemento que $\mathcal{E}$ no puede reconocer. Además, $\mathbb{MC}_{\mathcal{PBS}}$ no contiene ninguna información referente a la identidad de $\mathcal{C}$, gracias al uso de la firma parcial ciega.

5. `Pago Múltiple`. $\mathcal{C}$ puede pagar con cupones usando el protocolo de `Pago Múltiple` (Figura 4) a cualquiera de los $\mathcal{V}$ afiliados a $\mathcal{E}$. El protocolo de `Pago Múltiple` tiene cuatro pasos, mediante los cuales $\mathcal{C}$ puede gastar cualquier número de cupones con una sola ejecución del protocolo, incluso cupones pertenecientes a diferentes tiras de cupones. Esta característica no incluida en propuestas previas contribuye a mejorar la eficiencia de nuestra solución.

$\mathcal{C}$ firma usando el esquema de firma de grupo un conjunto de datos $(data_1)$ entre los cuales está la *información de pago* $(\mathbb{MC}_\omega^{pay})$. $\mathcal{V}$ valida la información y, si los datos recibidos son válidos ($\mathbb{MC}_\omega^{pay}$ no usado antes, $\mathbb{MC}_\omega^{pay}$ pertenece a $\mathbb{MC}_{\mathcal{PBS}}$, verificación de $\mathbb{MC}_{\mathcal{PBS}}$, etc.), envía un acuse de recibo juntamente con el servicio solicitado. A continuación, se repite el proceso, esta vez usando otro conjunto de datos $(data_2)$ análogo al anterior, pero conteniendo la *información de prueba* $\left(\mathbb{MC}_\omega^{proof}\right)$. Como antes, si las verificaciones son satisfactorias, $\mathcal{V}$ envía un acuse de recibo.

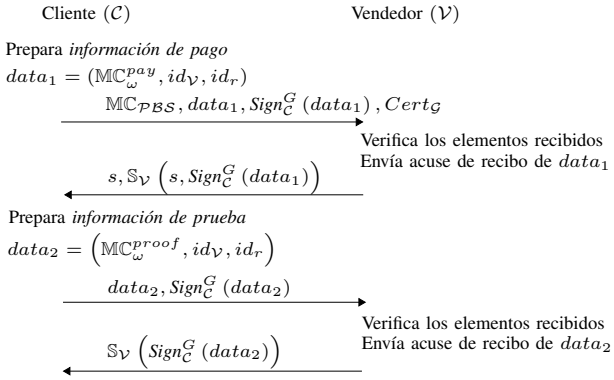


Figura 4: Protocolo de `Pago Múltiple`.

6. `Depósito` (*on-line* o *off-line*). El protocolo de `Depósito` permite a $\mathcal{V}$ solicitar a $\mathcal{E}$ un depósito correspondiente al valor de los cupones recibidos de los clientes. $\mathcal{V}$ puede depositar cupones por cada transacción de pago (`Depósito` *on-line*) o solo cuando tiene una lista de cupones (`Depósito` *off-line*).

7. `Reembolso`. En caso que $\mathcal{C}$ quiera recuperar el valor de cupones sin gastar, el protocolo le permite que $\mathcal{E}$ autorice su reembolso. Este protocolo es opcional y su aplicación dependerá de la implementación y del escenario.

IV. Comparación con una Propuesta Similar

A continuación comparamos la eficiencia de nuestra solución ($\mathcal{MC}-2\mathcal{D}$) frente al esquema multi-comerciante propuesto en [13]. Con el objetivo de comparar las medidas, hemos adaptado el escenario de pruebas a las condiciones de ejecución que los autores del esquema propuesto en [13] han realizado y analizado en [21] (un único portátil para cliente, comerciante y emisor, considerando la misma capacidad de cómputo). Además, hemos considerado las mismas pruebas

Cuadro I: Comparación práctica de rendimiento respecto a [13].

	Emisión (segundos)				**Pago Múltiple** (segundos)			
	$k = 5$ cupones			$k+1$	$k = 5$ cupones			$k+1$
	$\mathcal{C}$	$\mathcal{E}$	Total		$\mathcal{C}$	$\mathcal{V}$	Total	
[13]	-	-	4,280	0,811	-	-	33,01	6,476
Nuestra solución	0,023	1,182	1,205	$< 0{,}005$	0,877	1,204	2,082	$< 0{,}02$
Nuestra solución*	n/a	n/a	n/a	n/a	0,093	1,204	1,297	$< 0{,}02$

* - aplicando precomputación para la firma de grupo en el cliente durante el protocolo de `Pago Múltiple`
n/a - no aplicable

que en [21], teniendo en cuenta solo el tiempo de computación. De esta forma, comparamos el tiempo necesario para emitir y gastar un grupo de cinco cupones ($k = 5$) y la carga adicional de emitir o gastar un cupón adicional ($k + 1$). El Cuadro I recoge los resultados de rendimiento de ambas soluciones.

Analizando el proceso de emisión, el cliente solo necesita 23 *ms* de computación para obtener un $\mathbb{MC}^{2D}$, lo que significa que es aproximadamente 3,5 veces más rápido que el presentado en [21].

Referente al protocolo de `Pago Múltiple`, nuestra propuesta también obtiene mejores resultados. Tanto es así que el tiempo necesario para gastar 5 cupones es aproximadamente 15 veces menor que el mostrado en [21]. Si además aplicamos técnicas de precomputación para la firma de grupo, nuestro protocolo llegaría a ser hasta 25 veces más rápido, como se puede observar en el Cuadro I.

Finalmente, el tiempo necesario para emitir o gastar un cupón adicional en nuestro esquema, es despreciable. Esto es debido a que durante el protocolo de `Pago Múltiple`, $\mathcal{V}$ solo tiene que computar dos hashes para cada cupón adicional. Por tanto, aunque se gasten múltiples cupones durante una sola ejecución del protocolo, el tiempo de computación solo aumentará de forma lineal en función del coste de dos operaciones de hash por cada cupón. Como conclusión, el análisis demuestra que a diferencia de [13], $\mathcal{MC}-2\mathcal{D}$ es una solución escalable donde su rendimiento es independiente del número de cupones emitidos o gastados.

V. Evaluación del Rendimiento

Como hemos enfatizado en §I, no solo se debe tener en cuenta el coste computacional de las operaciones del protocolo, sino que también hay que introducir los costes debidos al efecto de la red. En esta Sección vamos a evaluar el rendimiento de $\mathcal{MC}-2\mathcal{D}$ utilizando un dispositivo Android como plataforma cliente. Además, hemos añadido la lógica necesaria para implementar la comunicación entre clientes y servidores remotos.

V-A. *Escenario de Pruebas*

La Figura 5 representa el escenario de pruebas considerado para obtener los valores de rendimiento de $\mathcal{MC}-2\mathcal{D}$. El escenario que proponemos emula un entorno real de producción con dispositivos móviles, servidores remotos y conexiones de red comerciales. Así pues, como plataforma de servidor, hemos elegido la solución Elastic Cloud Computing (EC2) de Amazon Web Services (AWS), para ejecutar el código

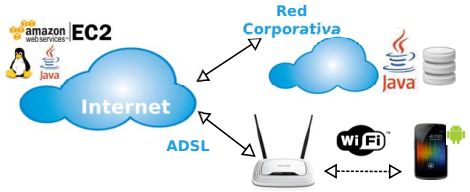


Figura 5: Escenario de pruebas.

del vendedor ($\mathcal{V}$), y un servidor virtual alojado en la red corporativa de la universidad, para ejecutar el código del emisor ($\mathcal{E}$). La aplicación cliente corre en un dispositivo Android, concretamente HTC Desire, que se conecta a los servidores remotos mediante una red WiFi y una conexión ADSL comercial. El Cuadro II resume las propiedades de los dispositivos considerados, mientras que el Cuadro III define las principales características de las dos redes consideradas.

Cuadro II: Dispositivos de test considerados.

Dispositivo	**Rol**	**CPU**	**RAM**	**OS**
Virtualbox	$\mathcal{E}$	2.8 GHz	1GiB	Debian Linux
EC2 μ-instance	$\mathcal{V}$	2 *EC2 CU*[(1)] ($\approx$ 1.0-1.2GHz)	633MiB	AWS Linux
HTC Desire	$\mathcal{C}$	1GHz	512MiB	Android 2.3

[(1)] Una *EC2 CU* (*Compute Unit*) proporciona la CPU equivalente a un procesador Xeon a 1.0-1.2GHz [25]

Cuadro III: Características de las redes.

Camino		**Tasa de transmisión (media)**		**Latencia (media)**
Origen	**Destino**	**Bajada**	**Subida**	**Round-trip**
$\mathcal{C}$ (ADSL)	$\mathcal{V}$	<3Mbps	<0.3Mbps	>200 ms
$\mathcal{V}$	$\mathcal{E}$	>25Mbps	>25Mbps	<100 ms

V-B. *Tiempo de Respuesta y Longitud de Mensajes*

Para conocer el rendimiento de la solución, hemos analizado el tiempo de respuesta total percibido por la aplicación cliente, realizando pruebas para los protocolos en las que está involucrada, es decir, los protocolos de `Registro`, `Emisión` y `Pago Múltiple`. Todas las pruebas se han repetido 20 veces y se ha realizado la media de los valores obtenidos descartando los resultados extremos.

En el tiempo de respuesta total percibido por el cliente, podemos distinguir dos factores principales:

- Tiempo de computación. La aplicación cliente tiene que realizar cálculos y operaciones matemáticas para generar las peticiones y procesar las respuestas recibidas. En este caso, los valores temporales dependen de la capacidad de procesamiento de cada dispositivo, principalmente de la CPU y de la memoria disponible.
- Tiempo de transmisión de red. El tiempo consumido por la aplicación para enviar y recibir datos a través de la red.

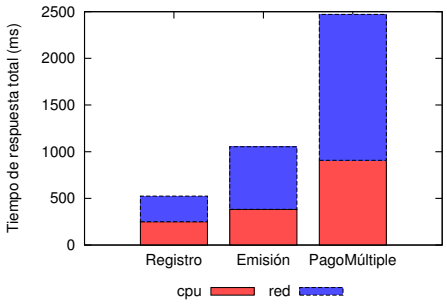


Figura 6: Tiempo de respuesta total para cada protocolo (usando HTC Desire y acceso WiFi).

La Figura 6 muestra el tiempo de respuesta medido en el cliente para cada uno de los protocolos analizados usando el dispositivo móvil HTC Desire. El protocolo que implica mayor coste es el de `Pago Múltiple`, dado que es el que incorpora mayor número de operaciones, tanto para cliente como para vendedor (entre las cuales hay que destacar dos firmas de grupo) y también un mayor tiempo de espera de red.

Si analizamos la Figura 7, podemos observar como la mayor parte del tiempo consumido por cada uno de los protocolos es básicamente debido a tareas de red: enviar mensajes y quedar en espera de recibir los mensajes de respuesta. Las tareas de red llegan a significar incluso más del 50 % del tiempo de respuesta total percibido por la aplicación cliente.

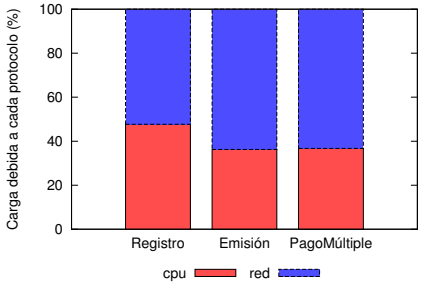


Figura 7: Porcentaje de carga debido a cada una de las tareas.

Si también analizamos el resultado obtenido teniendo en cuenta una ejecución *on-line* del protocolo de `Depósito` por parte del vendedor durante el `Pago Múltiple`, podemos afirmar que el tiempo añadido en el tiempo de total de respuesta es de solo 680 *ms*. Por consiguiente, el hecho de ejecutar una validación *on-line* de los cupones por cada ejecución del `Pago Múltiple`, representa un coste asumible en caso que se requiera comprobar inmediatamente con el emisor si los cupones recibidos no se han usado previamente.

En definitiva, además de evidenciar la eficiencia y el rendimiento de $\mathcal{MC}-2\mathcal{D}$, el análisis también demuestra que no solo es necesario evaluar el coste respecto al uso de los recursos de procesamiento necesarios, sino que también es imprescindible evaluar el tiempo consumido en la transferencia de datos a través de la red. Tanto es así que hemos demostrado que el tiempo necesario para enviar y recibir mensajes, puede ser incluso más importante que el tiempo de procesamiento de los mismos mensajes.

VI. Conclusiones

En este trabajo hemos comprobado como nuestra propuesta de cupones electrónicos para el escenario multi-comerciante es eficiente, escalable y que puede ser usada en dispositivos móviles. Para demostrarlo, hemos implementado la solución, comparado su eficiencia respecto a la propuesta previa y analizado su rendimiento considerando un escenario real. En primer lugar, la comparación nos permite afirmar que $\mathcal{MC}-2\mathcal{D}$ es más eficiente, utilizando el mismo escenario de pruebas que la propuesta anterior y considerando solo el efecto de la computación. Esto es debido, principalmente, a la utilización de mecanismos criptográficos con una menor carga y a la capacidad de nuestro esquema de permitir emitir y gastar cupones utilizando una misma transacción. En segundo término, la implementación completa del esquema sobre la plataforma Android nos ha permitido realizar una evaluación del rendimiento considerando un escenario realista, con servidores remotos y comunicaciones reales, escenario alejado de los entornos de prueba limitados que normalmente encontramos en las propuestas científicas. De esta manera, las medidas reflejan todos los factores que pueden afectar al tiempo de respuesta. De hecho, hemos podido comprobar que los costes debidos a otras tareas diferentes a la computación deben también ser analizados cuidadosamente para obtener una solución viable y con un tiempo de respuesta adecuado para el entorno de ejecución de la misma.

Como trabajo futuro, sería interesante estudiar la viabilidad de llevar a cabo el pago con cupones en los comercios utilizando tecnología NFC.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el Ministerio de Educación y Ciencia bajo el proyecto CONSOLIDER-ARES (CSD2007-00004).

Referencias

[1] BuddeComm. Global Mobile Communications - Statistics, Trends and Regional Insights. Report, verificado en Abril, 2014. http://goo.gl/Vw40qu.

[2] Techcrunch. Forrester 2012-2017: e-Commerce Forecast Analysis. Sitio web, verificado en Abril, 2014. http://goo.gl/7Dtwup.

[3] Edenred. Compañía internacional de talonarios de cupones para restaurantes. Sitio web, verificado en Abril, 2014. http://www.edenred.com.

[4] Gourmet. Compañía internacional de talonarios de cupones para restaurantes. Sitio web, verificado en Abril, 2014. http://www.cheque-dejeuner.com/.

[5] Bancotel. Compañía internacional de talonarios de cupones para hoteles. Sitio web, verificado en Abril, 2014. http://www.bancotel.es/.

[6] Boying Zhang, Jin Teng, Xiaole Bai, Zhimin Yang, and Dong Xuan. P3-coupon: A probabilistic system for Prompt and Privacy-preserving electronic coupon distribution. In *PerCom*, pages 93–101. IEEE, 2011.

[7] Carlo Blundo, Stelvio Cimato, and Annalisa De Bonis. Secure e-coupons. *Electronic Commerce Research*, 5:117–139, January 2005.

[8] Liqun Chen, Matthias Enzmann, Ahmad-Reza Sadeghi, Markus Schneider, and Michael Steiner. A Privacy-Protecting Coupon System. In *Financial Cryptography and Data Security*, volume 3570 of *Lecture Notes in Computer Science*, pages 578–578. Springer-Verlag, Berlin, Heidelberg, 2005.

[9] Sébastien Canard, Aline Gouget, and Emeline Hufschmitt. A handy multi-coupon system. In *Applied Cryptography and Network Security*, volume 3989 of *Lecture Notes in Computer Science*, pages 66–81. Springer-Verlag, Berlin, Heidelberg, 2006.

[10] Lan Nguyen. Privacy-protecting coupon system revisited. In *Financial Cryptography and Data Security*, volume 4107 of *Lecture Notes in Computer Science*, pages 266–280. Springer-Verlag, Berlin, Heidelberg, 2006.

[11] Liqun Chen, B. Alberto N. Escalante, Hans Löhr, Mark Manulis, and Ahmad-Reza Sadeghi. A privacy-protecting multi-coupon scheme with stronger protection against splitting. In *Proceedings of the 11th International Conference on Financial Cryptography and 1st International Conference on Usable Security*, volume 4886 of *Lecture Notes in Computer Science*, pages 29–44. Springer-Verlag, Berlin, Heidelberg, 2007.

[12] Alberto N. Escalante, Hans Löhr, and Ahmad-Reza Sadeghi. A non-sequential unsplittable privacy-protecting multi-coupon scheme. In *GI Jahrestagung (2)*, pages 184–188, 2007.

[13] Frederik Armknecht, B. Alberto N. Escalante, Hans Löhr, Mark Manulis, and Ahmad-Reza Sadeghi. Secure multi-coupons for federated environments: privacy-preserving and customer-friendly. In *Proceedings of the 4th International Conference on Information Security Practice and Experience*, volume 4991 of *Lecture Notes in Computer Science*, pages 29–44. Springer-Verlag, Berlin, Heidelberg, 2008.

[14] Liu Xin and Qiu liang Xu. Practical compact multi-coupon systems. In *IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, volume 3, pages 211 –216, 2009.

[15] Sue-Chen Hsueh and Jun-Ming Chen. Sharing secure m-coupons for peer-generated targeting via eWOM communications. *Electronic Commerce Research and Applications*, 9:283–293, July 2010.

[16] A. Pere Isern-Deya, M.F. Hinarejos, JL. Ferrer-Gomila, and M. Payeras-Capellà. A secure multicoupon solution for multi-merchant scenarios. In *IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pages 655–663, 2011.

[17] Xiaoling Dai and John Grundy. NetPay: An off-line, decentralized micro-payment system for thin-client applications. *Electronic Commerce Research and Applications*, 6(1):91–101, 2007.

[18] Jen-Ho Yang and Chin-Chen Chang. A low computational-cost electronic payment scheme for mobile commerce with large-scale mobile users. *Wireless Personal Communication*, 63(1):83–99, March 2012.

[19] Wenmin Li, Qiaoyan Wen, Qi Su, and Zhengping Jin. An efficient and secure mobile payment protocol for restricted connectivity scenarios in vehicular ad hoc network. *Computer Communications*, 35(2):188 – 195, 2012.

[20] Francisco Borrego-Jaraba, Pilar Castro Garrido, Gonzalo Cerruela García, Irene Luque Ruiz, and Miguel Ángel Gómez-Nieto. A Ubiquitous NFC Solution for the Development of Tailored Marketing Strategies Based on Discount Vouchers and Loyalty Cards. *Sensors*, 13(5):6334–6354, 2013.

[21] Alberto N. Escalante. Privacy-protecting multi-coupon schemes with stronger protection against splitting. In *Master's Thesis. Department of Computer Science. Saarland University*. 2008.

[22] David Chaum. Blind Signatures for Untraceable Payments. *Advances in Cryptology Proceedings of Crypto 82*, pages 199–203, 1983.

[23] Hung-Yu Chien, Jinn-Ke Jan, and Yuh-Min Tseng. RSA-based Partially Blind Signature with Low Computation. *International Conference on Parallel and Distributed Systems ICPADS 2001*, pages 385–389, 2001.

[24] Dan Boneh, Xavier Boyen, and Hovav Shacham. Short Group Signatures. In *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 227–242. Springer-Verlag, Berlin, Heidelberg, 2004.

[25] Amazon Web Services. Amazon Elastic Compute Cloud: Instance Types and Compute Resources Measurement, verificado en Abril, 2014. http://goo.gl/k0CJhn.

Análisis Visual del Comportamiento de Aplicaciones para Android

Oscar Somarriba, Ignacio Arenaza-Nuño, Roberto Uribeetxeberria, Urko Zurutuza

Dpto. de Electrónica e Informática
Mondragon Goi Eskola Politeknikoa
Mondragon Unibertsitatea
Emails: oscar.somarriba@alumni.mondragon.edu, {iarenaza,ruribeetxeberria,uzurutuza}@mondragon.edu

***Resumen*—Los dispositivos móviles inteligentes equipados con funciones avanzadas de computación y comunicaciones han crecido rápidamente. Sin embargo, a pesar de los mecanismos de seguridad existentes, y dada la cantidad creciente de dispositivos conectados a Internet, también han aumentado exponencialmente la cantidad de aplicaciones maliciosas (malware) dirigidas a ellos. En este trabajo mostramos cómo los componentes peligrosos y maliciosos del malware móvil se pueden visualizar de una manera intuitiva a fin de descubrir fácilmente qué funciones de Android pueden desencadenar el fraude. Nuestro enfoque incluye un método para interceptar llamadas a funciones ("hooking") con el fin de recoger trazas pertinentes de la aplicación durante de tiempo de ejecución. Esto permite la monitorización de llamadas a funciones de la API de Android relacionadas con los permisos de instalación de la misma. Las trazas obtenidas se colectan en un servidor web central donde tiene lugar la visualización del comportamiento de las aplicaciones.**

***Palabras clave*—seguridad en aplicaciones móviles (*mobile application security*), software malicioso en aplicaciones móviles (*mobile malware*), análisis Visual (*visual analytics*), análisis del comportamiento de aplicaciones móviles (*application behavior analysis*)**

I. Introducción

La adopción masiva de las comunicaciones móviles en la vida cotidiana ha traído una necesidad de establecer en la sociedad una confianza en la infraestructura móvil, y esto supone un gran reto en la actualidad. Esto es debido a que las plataformas móviles, como teléfonos inteligentes y tabletas, así como las aplicaciones móviles están aumentando exponencialmente en popularidad. Actualmente existen alrededor de 1 millión de aplicaciones para el Sistema Operativo móvil Android en su sitio web de ventas en línea *Google Play*, con un estimado de 50 mil millones de descargas [1]. En contraste, el software malicioso (malware) que ataca la plataforma Android ha aumentado considerablemente en los últimos 24 meses en un 100 %. Las nuevas familias de malware Android están evolucionando rápidamente para evitar ser detectados por los escáneres tradicionales basados en firmas. Hay una necesidad de mejorar las capacidades de detección para superar los nuevos desafíos de detección debido a la ofuscación, y así mitigar o remediar el impacto de la evolución de malware para Android.

Dado que el sistema operativo móvil más popular es Android OS de la compañia Google (en la actualidad tiene el 70 % del mercado), Android es el OS más atacado con un 99 % de los ataques malware según lo publicado por Cisco y Kapersky Labs durante el tercer trimestre Q3 del 2013, y resumido en el reporte de SOPHOS [2]. Esta investigación se enfoca en el seguimiento del comportamiento en tiempo de ejecución de las applicaciones y la visualización de sus funciones maliciosas para descubrir qué tipo de ataques o intenciones existen detrás de estas. La plataforma propuesta de monitorización está compuesta básicamente de cuatro elementos, a saber: (i) una aplicación Android llamada (*Sink*) que guía al usuario en la selección y parametrización de la aplicación a supervisar, (ii) un cliente embebido que se inserta en cada aplicación a ser supervisada, (iii) un servicio web encargado de recoger la aplicación a monitorizar, enviar al dispositivo la aplicación instrumentada, y recopilar las trazas que va generando, (iv) y finalmente un componente de visualización que muestra grafos relacionados con el comportamiento de las trazas o llamadas a funciones, relativa a la aplicación monitorizada.

Se prevé que esta herramienta pueda ser utilizada por analistas malware con el fin de realizar una inspección visual de las aplicaciones en estudio. Por otra parte, la monitorización de una aplicación en el momento de su ejecución es esencial para entender cómo esta interactúa con el dispositivo, con componentes claves tales como las APIs (*Application Programming Interfaces*) provistas por el sistema. Una API especifica cómo algunos componentes de software (rutinas, protocolos y herramientas) deben actuar cuando estén sujetos a invocaciones de otros componentes. Al rastrear y analizar estas interacciones, podemos ser capaces de dar seguimiento a cómo se comportan las aplicaciones, a cómo manejan datos sensibles e interactuar con el sistema operativo.

I-A. Contribución y organización del artículo

A lo largo de este artículo se presenta un método para la detección de malware, mediante el análisis visual de la ejecución de las funciones a las que llaman. La subsiguientes partes del trabajo están organizadas como a continuación se detalla. La sección II le da al lector las nociones básicas detrás de los componentes utilizados en las siguientes secciones y recopila el trabajo previo relacionado con la temática. La sección III describe la arquitectura de monitorización y visualización del sistema presentado. La sección IV muestra los resultados

obtenidos con la infraestructura implementada haciendo uso de diferentes aplicaciones. Por último, las secciones V y VI presentan las conclusiones e identifican unas posibles líneas futuras de trabajo, respectivamente.

II. Antecedentes y Trabajos relacionados

En los ambientes convencionales de Java, el código fuente es compilado en un conjunto de instrucciones llamado *byte-code*, el cuál es almacenado en un formato de ficheros .class. Estos ficheros son más tarde leídos por la Máquina Virtual de Java (JVM) al momento de su ejecución. Por otra parte, en Android, el código fuente de Java que ha sido compilado en ficheros .class debe ser convertido en ficheros .dex, frecuentemente referidos como ficheros ejecutables del tipo Dalvik (Dalvik Executable). Además, *Android application package file* (apk), es el formato de fichero utilizado para distribuir e instalar software de aplicaciones y middleware en el sistema operativo Android. Las apps se presentan en un fichero con el formato .apk, en un contenedor de la aplicación binaria que consiste en ficheros .dex, AndroidManifest.xml, y los ficheros de recursos de la aplicación. Asimismo, el archivo .apk resultante se firma con una clave (*keystore*) para establecer la identidad del autor de la app. Existen métodos para realizar el proceso en el sentido inverso. *Apktool* es un conjunto de herramientas para realizar ingeniería inversa en apps, lo que simplifica el proceso de ensamble y desensamble de ficheros binarios de Android (.apk) a ficheros Smali (.smali), permitiendo la modificación del código fuente. Esto resulta especialmente útil para el análisis de las aplicaciones.

El análisis y detección de malware para Android ha sido un tema candente de la investigación en los últimos años. Un ejemplo de los mecanismos de inspección para la identificación de aplicaciones con malware para Android se presenta en [3], donde también se desarrolló un sistema de instrumentación transparente para la automatización de las interacciones de los usuarios.

Además, en [4] se utiliza un marco de seguridad llamado *XManDroid* para extender el mecanismo de seguimiento de Android, con el fin de detectar y prevenir ataques del tipo escalada de privilegios a nivel de aplicación durante el tiempo de ejecución sobre la base de una política determinada. Adicionalmente, los autores en [5] y [6] han propuesto diferentes técnicas de seguridad con respecto a los permisos de las apps. Por ejemplo, en este último, se propone una herramienta para extraer la especificación de permisos del código fuente de Android OS. Por otra parte, las técnicas de detección de malware en dispositivos móviles usualmente se pueden clasificar de acuerdo al modo en el que se realiza el análisis: análisis estático y análisis dinámico. La primera se basa en intentar identificar el código malicioso por descompilación de la aplicación y la búsqueda de cadenas o bloques de códigos sospechosos; en la segunda se analiza el comportamiento de una determinada aplicación utilizando la información de su estado de ejecución. Algunos tipos recientes de detección de malware son: *Dendroid* [7] como un ejemplo de un análisis estático para dispositivos con Android, y *Crowdroid*, sistema que agrupa la frecuencia de llamadas al sistema de las aplicaciones para detectar malware [8]. En un reciente trabajo de Jiang y Zhou [9] se han mapeado los tipos más comunes de violaciones de permisos en un gran conjunto de datos de malware. Por otro lado, en [10], [11] se pueden encontrar estudios más amplios sobre el estado del arte de la seguridad para los dispositivos móviles.

Entre los sistemas de monitorización de comportamiento de aplicaciones se encuentra una propuestas que permite visualizar mediante grafos las llamadas a funciones de una aplicación determinada, pero los autores lo hacen mediante técnicas de análisis estático [12]. El sistema hace un mapa de todas las funciones disponibles, mientras que este trabajo solo monitoriza y visualiza aquellas funciones que se suceden en tiempo de ejecución, obteniendo además los parámetros que se envían a la función. No se han encontrado propuestas de detección de malware en base a análisis dinámico que opere en el dispositivo del usuario de manera muy ligera y ”online”. Esto es necesario debido a la apertura de la Plataforma Android donde el malware puede también ser instalado a través de apps de otras fuentes, tales como páginas web y de memorias USB, lo que requiere mecanismos de detección que operan en el propio dispositivo.

III. Descripción del sistema

En esta sección se presenta una solución aplicable para la detección de anomalías producto de la presencia de malware en las aplicaciones, basada en un análisis dinámico combinado con el soporte de un servidor web.

La Figura 1 muestra la estructura de la plataforma de monitorización propuesta. Las etapas para llevar a cabo la misma consisten en los siguientes pasos lógicos: Etapa I: Envió de la aplicación **APP** y de una lista de permisos que se desean monitorizar al Servidor Web, Etapa II: Instrumentación de la aplicación mediante un proceso de *hooking* generando **APP'**, Etapa III: Instalación y activación de la **APP'** reemplazando a **APP** en el dispositivo, Etapa IV: Almacenamiento de las trazas de **APP'** en una base de datos, y la Etapa V: Visualización de los grafos relacionados con **APP'**.

De nuevo, en la Figura 1 se muestra un diagrama de bloques formado por cuatro componentes: la aplicación *Sink*, un cliente embebido, un servidor web, y el componente de visualización.

III-1. La Aplicación Sink: es una aplicación Android con dos funciones principales: una de gestión de la aplicación a monitorizar, y otra para el manejo de las trazas. La parte del tratamiento de la aplicación, a su vez, está compuesta de un conjunto de actividades como se muestra en la Figura 2(d).

En la Figura 2(a) el usuario selecciona la aplicación que desea monitorizar, entre aquellas que no vienen reinstaladas de fábrica. En el siguiente paso se seleccionan los permisos que el usuario estime convenientes a monitorizar, relacionados con la aplicación y considerados como peligrosas según el mapa de funciones API obtenido con PScout [6]. La interfaz guía después al usuario a lo largo de varias actividades donde se llevan a cabo la subida de la aplicación y lista de permisos a monitorizar al Servidor, descarga de la aplicación modificada,

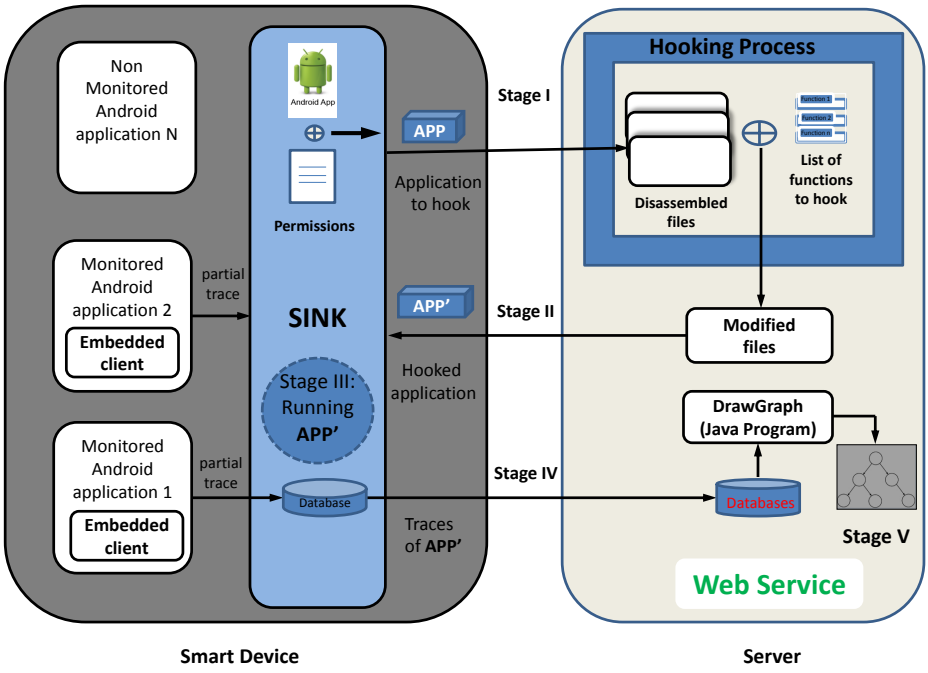


Figura 1: Componentes del Sistema de Monitorización del comportamiento de la Aplicación.

e instalación de la misma. Finalmente, un botón permite iniciar o detener la monitorización de la aplicación en el cualquier instante.

Por otra parte, el Sink permite realizar la gestión de las trazas, ejecutando servicios en segundo plano. Este gestor es el encargado de colectar las trazas enviadas desde los clientes embebidos individuales, ubicados en cada una de las aplicaciones supervisadas, añadiéndoles una marca de tiempo y el hash del ID del dispositivo, y su almacenamiento en una memoria intermedia circular común. Por último, las trazas se envían periódicamente al servicio web donde se almacenan en una base de datos.

III-2. El Cliente Embebido: consiste en un módulo de comunicación que utiliza el protocolo UDP para la transmisión de las trazas de las funciones modificadas invocadas en **APP'**, al Sink.

III-3. El Servicio Web: provee los siguientes servicios al *Sink*: subir aplicaciones, descargar las aplicaciones modificadas y enviar las trazas. Ahora la pieza clave de todo el sistema y donde reside la lógica del método presentado, es la herramienta que instrumenta la aplicación, el proceso conocido como "hooking". Este componente mapea los permisos de la aplicación en llamadas a funciones que son marcadas, las cuáles van ser monitorizadas. Por lo tanto, este proceso es una acción automática realizada por el servidor Web cada vez que una aplicación es enviada al mismo.

Las funciones modificadas registrarán el *nombre de la aplicación*, el nombre de paquete de la aplicación, y el hash de la aplicación y enviará esta información al cliente embebido junto con el nombre de la función (e.g., *sendTextMessage()*, *getDeviceID()*, *execSQL()*, *SendBroadcast()*, etc.). A continuación, todos los ficheros modificados junto con el resto de los recursos desensamblados se reensamblan y se empaquetan en un fichero binario Android .apk. Al terminar la Etapa II, la **APP'** es descargada al *Sink*.

III-4. Visualizaciones: Con el fin de realizar un análisis visual del comportamiento de las aplicaciones se utiliza una base de datos NoSQL basada en grafos, Neo4j[1]. Neo4j almacena los datos en una estructura orientada a grafos, en lugar de utilizar las tablas relacionales de las bases de datos convencionales. En términos generales, un grafo es una representación de un conjunto de nodos y las relaciones entre ellos unidos por medio de enlaces, (vértices y aristas o arcos, respectivamente). Esto se ilustra en la *Etapa V* de la Figura 1. De esta manera, se puede plasmar cada uno de los comportamientos de la aplicación analizada con una representación simple pero muy ilustrativa. Los grafos se elaboran mediante relaciones de tipo "*una Aplicación incluye varias Clases que a su vez llaman a Funciones*". El primer nodo superior, "Aplicación", contiene el nombre del paquete de la aplicación, que es única para cada una de las aplicaciones existentes, mientras que el segundo nodo, "Clase", representa el nombre del componente de Android que ha llamado a la "API call", el nodo "Función".

Una vez se obtiene la información recogida por el servicio Web en la base de datos, toda su estructura de llamadas a funciones puede ser filtrada y tratada. Inicialmente se genera un grafo sin incluir colores empleando el lenguaje *Cypher Query*

[1]Software disponible en el sitio web http://www.neo4j.org

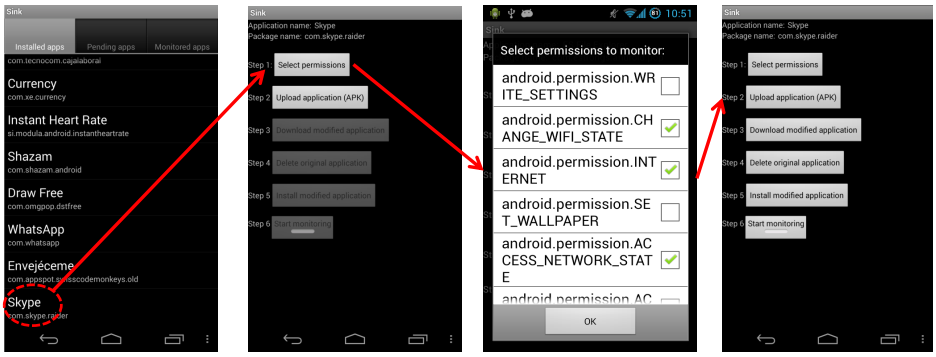


Figura 2: Interfaz de Usuario del *Sink*. (a) Selección de la aplicación, (b) Pasos de la aplicación, (c) Selección de los permisos, y (d) Proceso de monitorización.

Languaje, que permite operar y aplicar transformaciones en el grafo.

Para ello, un experto debe completar y encadenar un conjunto de reglas que ayudan a resaltar el comportamiento malicioso conocido (por ejemplo, la llamada a la función *SendMessageText()*. Para ello, se buscan los nodos en la parte inferior del grafo relacionado con las llamadas a funciones API consideradas maliciosas. Las reglas incluyen la búsqueda y representación de funciones maliciosas (representadas en rojo), sospechosas o maliciosas pero no críticas (en naranja) y benignas (en verde). Cuando se detecta un comportamiento malicioso como el envío de mensajes SMS a un número de pago premium sin el consentimiento del usuario, se procede a representar el nodo bajo inspección en color rojo como señal de alerta usando Cypher.

IV. Resultados

En esta sección se muestran los resultados de utilizar la plataforma de monitorización y visualización para algunos ejemplos. En este artículo, exploramos 3 diferentes apps con el fin de evaluar todo el marco de trabajo:

- **Skype-free IM & video calls**
- La aplicación popular **Angry Birds**
- **El malware Fake player**

IV-A. Las Trazas

Después de ejecutar las aplicaciones arriba mencionadas durante 2-3 de minutos cubriendo todas las funcionalidad de la aplicación, el servidor Web recolecta un gran volumen de trazas de cada una de las mismas.

IV-B. Análisis Visual de las Trazas

Como se ha dicho anteriormente, un conjunto de reglas predefinidas por expertos nos permite identificar las funciones API ”sospechosas”, y en función de sus parámetros, se asignan colores a éstas. Al hacerlo, nos permite identificar rápidamente las funciones y asociarla con elementos relacionados. Al aplicar la clasificación de funciones en base a un color para cada nodo del grafo, esto permite la construcción de un ”mapa visual”que describe y ayuda al análisis de su funcionamiento. Además, este grafo es adecuado para guiar el analista durante el examen de clasificación de una muestra de malware peligrosa debido a que el sombreado rojo de los nodos indican estructuras maliciosos identificados por la infraestructura de monitorización. Esta revisión debe realizarse entre todos los nodos de las funciones llamadas en el nivel más bajos de cada rama del árbol del grafo. Sin embargo, con el fin de colorear completamente el grafo de la aplicación hasta llegar al nodo raíz, hay que recurrir a realizar un análisis de abajo hacia arriba (”bottom-top”) del vecindario de cada función invocada y las asociadas. Por lo tanto, si una de las ramas del grafo es coloreada en rojo, a continuación, la app se considera como potencialmente maliciosa.

En particular, los grafos de las aplicaciones como *Skype*, *Angry Birds*, y *Fake Player* se muestran en la Fig. 3 lo cual proporciona al usuario una indicación del estado de seguridad de éllos. Para ello, se han creado reglas Cypher para colorear aquellos nodos que contienen una llamada a función de envío de SMS en rojo, y llamadas a funciones para obtener y mostrar publicidad (Adware) en naranja.

Como resultado, el grafo generado para la aplicación *Skype* no muestra ninguna amenaza y sus nodos aparecen coloreados en verde, tal y como se muestra en la Fig. 3(a).

Por otra parte, en la Fig. 3(b) existe una aplicación con Adware, por lo que varios nodos de esta aplicación están coloreados en naranja, mientras en contraste las funciones maliciosas que identifican un malware se colorean en rojo,

como se muestra en la Figura 3(c) para la aplicación Fake Player.

V. Conclusiones

En este trabajo se propone una arquitectura de supervisión con el objetivo de monitorizar aplicaciones Android a gran escala, sin modificar el firmware del mismo, sin la necesidad de obtener permisos de administrador del dispositivo, y que resulta en un grafo de visualización donde se destacan las llamadas a función correspondientes a los comportamientos de malware predefinido. La plataforma está compuesta por cuatro componentes: el cliente embebido, el Sink, el servicio web y la visualización. Antes de que una aplicacón sea supervisada, el Sink la transfiere al Servicio Web, que se encarga de la inserción de los *hooks* añadiendo el cliente embebido en el interior la aplicación. Finalmente el *Sink* descargará la aplicación recién instrumentada. Cuando una función modificada es llamada, se construye una traza parcial que será pasada al cliente embebido que a su vez la enviará al *Sink*. Este recoge los trazas parciales de todas las aplicaciones supervisadas, las completa, y las sube mediante el servicio Web. El Servidor finalmente transforma las trazas y las almacena en una base de datos de grafos.

Por último, se aplican un conjunto de reglas predefinidas con el fin de obtener una visualización donde se pone de relieve o resalta la conducta maliciosa de la aplicación supervisada. La infraestructura desarrollada es capaz de monitorizar simultáneamente varias aplicaciones en distintos dispositivos y la recopilación de todos las trazas se da en un mismo lugar. Las pruebas realizadas en este trabajo muestran que las aplicaciones pueden ser preparadas para ser supervisadas en cuestión de minutos y las aplicaciones modificadas se comportan como estaban originalmente diseñadas. Además, se ha mostrado que la infraestructura se puede utilizar para detectar comportamientos maliciosos en aplicaciones, tales como el monitorizado del malware *Fake Player*. Las evaluaciones del *Sink* han revelado que nuestro sistema de supervisión es reactivo, no pierde ninguna de las trazas parciales, y tiene un impacto muy pequeño en el rendimiento de las aplicaciones supervisadas.

VI. Trabajos futuros

Como trabajo futuro, la plataforma se puede ampliar para ser capaz de supervisar las funciones Android conocidas como *Intents*, enviadas por la aplicación que permitirían a una aplicación llamar a funciones que no requieren ningún tipo de permiso específico (como por ejemplo que una función llame a un navegador sin que la aplicación tenga permisos de Internet). No ser capaz de monitorear *Intents* significa que la infraestructura no es capaz de realizar un seguimiento, de si la aplicación supervisada inicia otra aplicación durante un corto período de tiempo para realizar una tarea determinada, e.g., para abrir un navegador web para mostrar la EULA (end-user license agreement). Además, esto permitiría saber cómo la aplicación bajo prueba se comunica con el resto de las aplicaciones de terceras partes y las aplicaciones instaladas en el dispositivo.

Asimismo se puede ampliar el modelo anti-malware descrito en este trabajo, desarrollando una arquitectura que lo complemente mediante la detección automática de malware móvil como por ejemplo con el uso de VirusTotal, realizando así un paso de filtrado previo. En definitiva, se podría obtener un sistema mas versátil disponiendo en el móvil de una aplicación que reporte anomalías (i.e., desviación del patrón de tráfico de las aplicaciones de red) a un servidor anti-malware en su red.

Referencias

[1] Businessinsider, "businessinsider." Available Online, 2014.

[2] Sophos, "Sophos mobilebile security threat report." Available Online, 2014. http://www.sophos.com/en-us/medialibrary/PDFs/other/sophos-mobile-security-threat-report.ashx.

[3] M. Karami, M. Elsabagh, P. Najafiborazjani, and A. Stavrou, "Behavioral analysis of android applications using automated instrumentation," in *Software Security and Reliability-Companion (SERE-C), 2013 IEEE 7th International Conference on*, pp. 182–187, June 2013.

[4] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, and A.-R. Sadeghi, "Xmandroid: A new android evolution to mitigate privilege escalation attacks," Technical Report TR-2011-04, Technische Universität Darmstadt, Apr. 2011.

[5] J. Jeon, K. K. Micinski, J. A. Vaughan, A. Fogel, N. Reddy, J. S. Foster, and T. Millstein, "Dr. android and mr. hide: Fine-grained permissions in android applications," in *Proceedings of the Second ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '12, (New York, NY, USA), pp. 3–14, ACM, 2012.

[6] K. W. Y. Au, Y. F. Zhou, Z. Huang, and D. Lie, "Pscout: Analyzing the android permission specification," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, (New York, NY, USA), pp. 217–228, ACM, 2012.

[7] G. Suarez-Tangil, J. E. Tapiador, P. Peris-Lopez, and J. Blasco, "Dendroid: A text mining approach to analyzing and classifying code structures in android malware families," *Expert Syst. Appl.*, vol. 41, pp. 1104–1117, Mar. 2014.

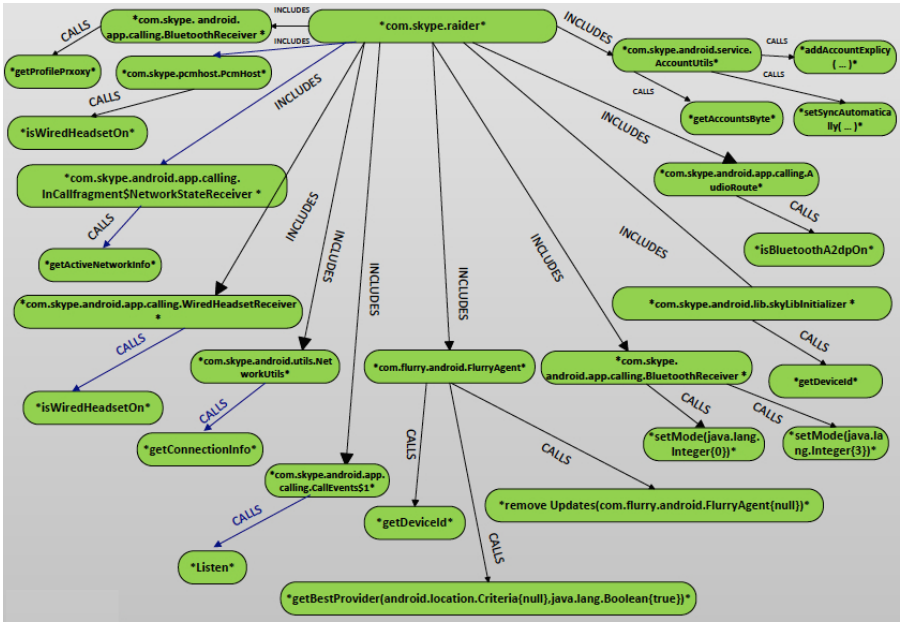
[8] I. Burguera, U. Zurutuza, and S. Nadjm-Tehrani, "Crowdroid: Behavior-based malware detection system for android," in *Proceedings of the 1st ACM Workshop on Security and Privacy in Smartphones and Mobile Devices*, SPSM '11, (New York, NY, USA), pp. 15–26, ACM, 2011.

[9] X. Jiang and Y. Zhou, *Android Malware*. Springer New York, 2013.

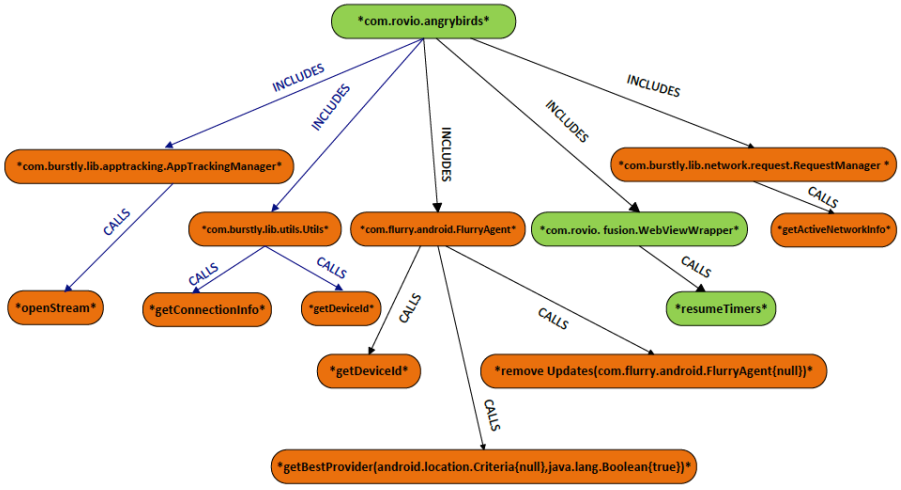
[10] M. La Polla, F. Martinelli, and D. Sgandurra, "A survey on security for mobile devices," *Communications Surveys Tutorials, IEEE*, vol. 15, pp. 446–471, First 2013.

[11] G. Suarez-Tangil, J. Tapiador, P. Peris-Lopez, and A. Ribagorda, "Evolution, detection and analysis of malware for smart devices," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–27, 2013.

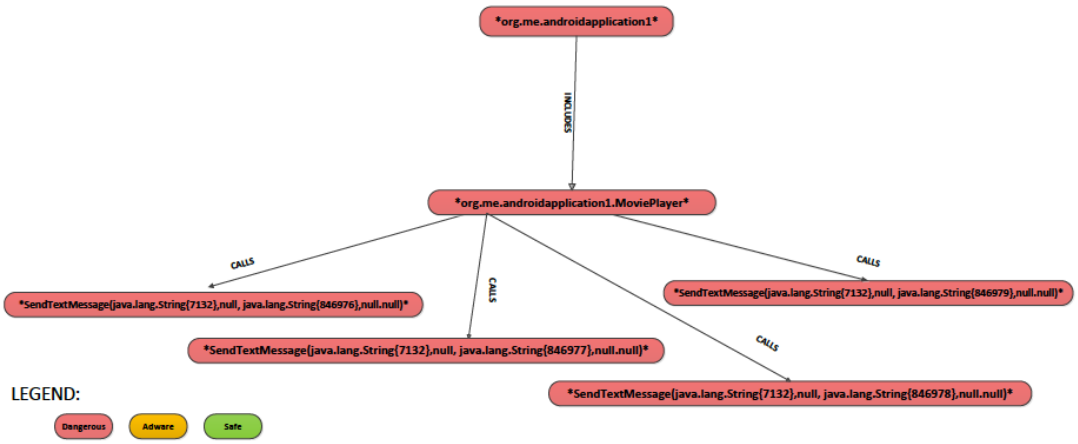
[12] H. Gascon, F. Yamaguchi, D. Arp, and K. Rieck, "Structural detection of android malware using embedded call graphs," in *Proceedings of the 2013 ACM workshop on Artificial intelligence and security*, pp. 45–54, ACM, 2013.



(a) Grafo de Skype.



(b) Grafo del juego Angry Birds.



(c) Grafo del malware Fake Player.

Figura 3: Grafos de las aplicaciones bajo prueba. (a) Diagrama Superior: Grafo de Skype, (b) Diagrama Intermedio: Grafo de la aplicación Angry Birds, y (c) Diagrama Inferior: Grafo del malware Fake Player.

Estudio práctico de mecanismos de seguridad en dispositivos Android

Enric Jódar Ciurana
Departament d'Enginyeria Telemàtica
Universitat Politècnica de Catalunya
Email: enric.jodar@entel.upc.edu

Josep Pegueroles Vallés
Departament d'Enginyeria Telemàtica
Universitat Politècnica de Catalunya
Email: josep.pegueroles@upc.edu

Juan Vera del Campo
Departament d'Enginyeria Telemàtica
Universitat Politècnica de Catalunya
Email: juanvi@entel.upc.edu

***Resumen*—Android climbed up to 80 percent of the smartphone and mobile devices market share. One of the key aspects for the acceptance of a mobile OS is the security degree the user perceives from the system. In this article, we explore some of the important security mechanisms implemented in Google Android through the study of several recent vulnerabilities. Particularly, we discuss a recent security issue in WhatsApp, the dangers of connecting devices to external machines and the security of current mechanisms for access control. We describe these vulnerabilities through in-lab proof-of-concepts. The experience learned from these cases is used to propose better practices for improving the security of the system.**

Palabras clave*—Android, acceso *(access control)*, cifrado *(encryption)*, revisión *(survey)*, seguridad *(security)

I. Introducción

El sistema operativo (SO) *Android*, presente tanto en *smarpthones* como en tabletas, ha sufrido un gran crecimiento en el número de usuarios provocando también que la comunidad de desarrolladores haya crecido [1]. De la misma forma las posibilidades para utilizar la plataforma *Android* con fines técnicos-comerciales también han crecido, se hace necesario realizar un estudio por parte de la comunidad sobre qué nivel de seguridad se puede garantizar en los dispositivos actuales.

Las prestaciones del primer dispositivo, el *HTC Dream* más conocido como *T-Mobile G1*, no tienen nada que ver con las prestaciones de los terminales actuales. Así pues, es razonable pensar que tanto las funcionalidades disponibles para el usuario como las herramientas del sistema hayan mejorado de forma considerable, permitiendo una multitud de aplicaciones que a su vez conllevan implicaciones a nivel de privacidad y seguridad de los datos del usuario.

Android está construido sobre el kernel de *Linux* y su código es de tipo abierto o *open-source*. Esto ha permitido identificar vulnerabilidades del sistema que posteriormente han sido subsanadas, ya sea a partir de acciones de los usuarios o bien por la determinación de Google para mejorar su sistema.

La falta de documentación técnica actualizada sobre dispositivos *Android* relacionada con la seguridad y el análisis de ésta es el motivo por el cual hemos decidido realizar este artículo. El análisis se ha hecho con ejemplos reales y recientes de algunas vulnerabilidades del sistema.

El artículo está organizado de la siguiente manera. La sección II hace referencia a anteriores artículos donde se revisa el estado del arte de la seguridad en dispositivos *Android* así como la percepción sobre el nivel de seguridad por parte de los usuarios. En la sección III se han estudiado los principales elementos de seguridad sobre los que va a tratar el artículo. En el apartado IV desarrollamos en detalle el análisis de seguridad mediante ejemplos concretos y en la sección V se hacen diversas propuestas que permitan mejorar la seguridad de nuestro dispositivo. Finalmente, este artículo acaba con las conclusiones de nuestro trabajo y referencias para consultar.

II. Trabajo relacionado

La elaboración de este artículo parte de un análisis realizado en el año 2009 [2] en el que se destacan los principales mecanismos de seguridad en el sistema operativo Android, así como se describen algunos cambios y propuestas concretas que permitirían mejorar en este aspecto. Aunque en el análisis realizado se hace referencia al móvil *HTC Dream*, cuyas prestaciones no se asemejan a la de los móviles actuales, el artículo nos han permitido obtener una visión más concreta sobre la construcción del sistema.

En [3] hemos podido analizar una encuesta hecha a 60 usuarios donde se estudia comparativamente las acciones que el usuario realiza en un ordenador personal frente a las que realiza en un *smartphone*. En el artículo se comprueba que los usuarios prefieren utilizar el ordenador al realizar operaciones con datos sensibles como por ejemplo introducir el número de la seguridad social, el número de cuenta bancaria, efectuar procesos de compra o bien intercambiar información personal relativa a la salud. El artículo muestra que un 60 % de los usuarios reticentes a utilizar el *smartphone* para realizar este tipo de operaciones argumentan motivos relacionados con la seguridad del dispositivo. En el estudio también se observan los criterios seguidos por parte de los usuarios al instalar aplicaciones *Android*. Los criterios más valorados son el precio, la popularidad y las críticas recibidas por otros usuarios. Sin embargo, los permisos, la política de privacidad y las condiciones de uso de la aplicación son los criterios que menos se valoran por parte de los usuarios.

III. Descripción del sistema Android

En esta sección se describen los principales mecanismos que intervienen en la seguridad de los dispositivos.

III-A. Mecanismos de cifrado

Android desde la versión 2.3.4 tiene soporte para el cifrado del sistema de ficheros aunque ésta opción se ofreció a los usuarios a partir de la versión 3.0, para ello se utiliza *dm-crypt*, herramienta presente en el *kernel*. El cifrado, según se describe en [4], se realiza con un algoritmo AES de 128 bits con el modo CBC. La *master key* se cifra con otra clave AES de 128 bits, para ello se utiliza la función PBKDF2, implementada en *OpenSSL*. La *sal* se genera a partir de una secuencia de números o contraseña establecida por el usuario.

Éste es un proceso irreversible en el cual se cifra por completo el sistema de ficheros, sólo se puede revertir en caso de realizar un borrado al estado de fábrica, perdiendo así los datos almacenados. A nivel lógico se sigue utilizando el sistema de ficheros de manera transparente para el usuario. Sin embargo, a nivel físico los datos se encuentran cifrados y no podrían ser extraídos por otro dispositivo sin la clave correspondiente. Cuando es necesario acceder a un fichero determinado éste se descifra y se vuelve a cifrar una vez finalizada su modificación. El proceso se realiza a nivel de bloque del sistema de ficheros.

Actualmente, el sistema también permite el cifrado de dispositivos de almacenamiento externo. Una vez realizado el proceso, los ficheros cifrados sólo serán accesibles desde el mismo dispositivo.

III-B. Firma de aplicaciones y Keystore

Cualquier aplicación debe estar firmada para poder ser instalada en el entorno *Android*. La firma del paquete (*APK*) se realiza a través de un certificado digital auto firmado, éste es generado a partir de la *keystore* creada por el desarrollador. Toda modificación y/o actualización del *APK* deberá firmarse con el mismo certificado. La utilización de autoridades de certificación se limita en el uso de la navegación segura y durante el uso de las *VPN* configuradas en el dispositivo.

A través de la *KeyStore* el sistema operativo realiza la gestión de claves criptográficas, de esta manera se facilita el almacenamiento de claves de forma segura por parte de las aplicaciones sin tener que aplicar medidas de seguridad adicionales durante el desarrollo.

En la versión 4.3 se han aplicado mejoras de seguridad debido a la capacidad de gestión multiusuario de las claves y la mejora en el respaldo de claves basado en sistemas *hardware* [5]. Así pues en los dispositivos multiusuario, una misma aplicación puede almacenar y gestionar diferentes claves dependiendo del usuario que utiliza la aplicación. Por otra parte, el respaldo de claves basado en sistemas *hardware* ofrece mayor seguridad ya que las claves no pueden ser exportadas ni manipuladas por ningún otro elemento que no sea el *hardware* usado para tal finalidad.

III-C. Mecanismos de control de acceso

Los dispositivos siempre han dispuesto de mecanismos de control de acceso. Ya en la versión *Gingerbread 2.2*, la segunda versión más utilizada después de *Jelly Bean* [6], la protección se realizaba a partir de un patrón, un *PIN* o una contraseña elegida por el usuario. Actualmente las opciones se han incrementado añadiendo la posibilidad de proteger el dispositivo mediante desbloqueo facial-voz o bien deslizando el dedo por la pantalla, aunque las últimas opciones contemplan un nivel muy bajo de seguridad.

Respecto al control de procesos y ficheros en el sistema, la ejecución de las aplicaciones se realiza de manera aislada debido al sistema *POSIX*, así pues cada paquete (*APK*) tiene asignado un *UserID (UID)* y el código de la aplicación se ejecuta en un proceso de manera aislada a la de cualquier otra. El *UID* también restringe el acceso a los archivos de la propia aplicación frente a otras siempre que éstos se almacenen en el directorio de la aplicación. El acceso al sistema de ficheros sigue las directrices establecidas en *Linux* (*rwx*) de la misma forma que ocurre con el sistema *POSIX*.

III-D. Sistema de permisos

En el caso de *Android*, durante la instalación de la aplicación se informa al usuario de los permisos que ésta va a requerir para su correcto funcionamiento. El usuario tiene la opción de aceptar el procedimiento o bien rechazarlo si cree que alguno de los permisos puede ser perjudicial para el dispositivo o los datos que éste contiene. La desventaja principal se encuentra en el hecho que no se puede aceptar o rechazar un subgrupo determinado de permisos.

III-E. Repositorio de aplicaciones (Play Store)

La tienda oficial de aplicaciones de *Android*, antiguamente conocida como *Google Play*, aún tiene fallos de seguridad importantes debidos a la falta de revisión de las *Apps* subidas por los desarrolladores. Recientemente Google ha publicado una patente que podría ser utilizada para evitar que aplicaciones *malware*, copias de otras ya existentes, sean introducidas en la tienda oficial [7]. De esta manera, cuando un desarrollador suba una aplicación, el sistema hará una comparación de los recursos utilizados con otras aplicaciones disponibles en la *Play Store*. En caso que un número razonable de recursos coincidan (ficheros multimedia, de datos o ejecutables), la aplicación pasará a ser revisada manualmente para comprobar si se trata de una aplicación pirata o copia de otra. Esta novedad permitirá ofrecer más seguridad tanto a los desarrolladores de las aplicaciones que pueden ser víctimas de copias o suplantación, como a los usuarios que podrán adquirir más confianza en la tienda oficial de *Android*.

III-F. Conectividad USB

Los móviles y las tabletas pueden conectarse mediante un *USB*, ya sea para copiar datos de usuario no protegidos o bien para realizar operaciones a través del *Android Debug Brigde (ADB)*. A partir de la versión 4.2, la opción para habilitar la conexión *USB* solo es visible en caso de activar el modo desarrollador en el dispositivo. Más segura es la conexión con la versión 4.2.2 ya que además de habilitar la conexión *USB*, se debe confirmar la conexión mediante la aceptación de la firma RSA del ordenador al cual conectamos nuestro dispositivo. A primera instancia la conexión *USB* puede parecer *inofensiva* y

las medidas de seguridad adoptadas en la versión 4.2.2 pueden parecer poco resolutivas pero más adelante comprobaremos que no es así.

IV. Análisis de seguridad

En esta sección se presentan tres vulnerabilidades que pueden o podrían haber afectado a una cantidad considerable de usuarios de *smartphones* o tabletas con consecuencias graves para la privacidad y seguridad de los datos de dichos usuarios.

IV-A. Vulnerabilidad en la privacidad de WhatsApp

En marzo de 2014 se conoció una importante vulnerabilidad que afecta a la privacidad de los usuarios del servicio *WhatsApp* [8]. La brecha de seguridad fue detectada al observar que el histórico de las conversaciones era guardado en una parte de la memoria interna (*/sdcard*) o en la *SD Card* externa, donde cualquier aplicación con permisos para acceder al almacenamiento externo y a Internet podía subir el histórico a un servidor remoto. Aunque el archivo de *backup* estaba cifrado, el método de cifrado era muy simple ya que utilizaba la misma clave de 192 bits en todos los dispositivos que utilizaban la aplicación. Esa clave fue descubierta y distribuida en 2011.

Posteriormente a la publicación de esta vulnerabilidad, *WhatsApp* cambió el método de cifrado y utilizó una clave única por dispositivo generada a partir de la cuenta *WhatsApp* asociada a éste. Aunque se produjo una mejora considerable, sólo con acceder a las cuentas del dispositivo (permiso *GETACCOUNTS*) y utilizar un nuevo método de descifrado, se pudo obtener nuevamente la clave [9].

Esta vulnerabilidad fue aprovechada en 2013 por una aplicación llamada *Balloon Pop 2* que se distribuyó a través de la *Play Store*, una vez detectada fue retirada por parte de *Google*. La aplicación, a grandes rasgos perseguía los mismos objetivos, las copias de las conversaciones eran almacenadas en un sitio web en el cual pagando una cantidad determinada y introduciendo el móvil de la víctima, cualquier usuario podía espiar el histórico de conversaciones.

IV-B. Conectividad USB

En octubre de 2013 fue reportado un fallo de seguridad relativo al mecanismo de control de acceso al dispositivo. Según el informe [10] la vulnerabilidad afectaba las versiones 4.0, 4.1, 4.2 y 4.3, posteriormente se depuró en la versión 4.4. En el informe se demuestra como cualquier dispositivo que tenga activada la depuración *USB* está expuesto a la amenaza. El fallo se encuentra en la implementación de la clase *ChooseLockGeneric*, ésta se ocupa de seleccionar el método de acceso al dispositivo por parte del usuario, en caso de existir uno y querer cambiarlo se debe introducir correctamente el anterior. Este error ha sido verificado en un dispositivo *Android 4.1.2* a través del comando *ADB* y la aplicación de test que proporciona el creador del informe y que permite inhabilitar el sistema de *login*. A su vez se ha comprobado que la vulnerabilidad no afecta a un dispositivo con *Android 2.3.7*, así pues suponemos que en alguna de las actualizaciones posteriores se debió introducir el error.

Otra amenaza a la cual podríamos estar sometidos los usuarios son los cargadores de batería de uso público. El hecho de disponer de un cargador público, ya sea con un coste para el usuario o bien de uso gratuito, es cada vez más frecuente debido a la poca duración de la batería de los dispositivos. En el campus universitario *Campus Nord (UPC)* disponemos de un equipamiento que suministra energía mediante la conexión de un cable *USB* de datos, como se muestra en la figura 1. En este equipamiento, la energía se obtiene a través de la radiación solar y se almacena en una batería conectada a la celda solar. Además, dispone de un conjunto de conectores dependiendo del dispositivo que se quiera conectar, el tiempo de conexión recomendado es de 30 minutos. Esta solución puede convertirse a su vez en una amenaza en caso que hubiera algún tipo de mecanismo , por ejemplo una *Raspberry* o cualquier elemento similar, que acceda a los datos del dispositivo e incluso pueda copiarlos. En este caso, la extracción de la información se podría llevar a cabo a través de un script que ejecute comandos en *ADB* y realizar así una copia de determinada información. Al utilizar esta infraestructura, el usuario debe confiar en el buen uso que se hace de ella por parte de la empresa o institución responsable, aunque cabe recordar que si este elemento se encuentra en el espacio público, podría ser incluso manipulado.

IV-C. Control de acceso: reconocimiento facial

Por último, hemos realizado un *test* para comprobar la seguridad del mecanismo de acceso mediante reconocimiento facial, técnica implementada en la versión 4.0 de la plataforma. Para ello hemos habilitado el mecanismo en la *Samsung Galaxy Tab3* y posteriormente hemos realizado una fotografía con una tableta *Sony Xperia Z* a la misma persona que ha activado el reconocimiento facial. Se ha podido comprobar que se ofrece un nivel de seguridad bajo, tal como se indica en la subsección III-C, ya que el dispositivo protegido por reconocimiento facial ha sido desbloqueado utilizando la fotografía realizada en la tableta *Sony Xperia Z*. Cualquier persona que tenga acceso a nuestro dispositivo y a una foto nuestra, ya sea impresa en papel o mostrada por pantalla, podría conseguir introducirse en nuestro sistema. Para incrementar la seguridad del mecanismo, en la versión 4.1 se añadió la necesidad de parpadear durante el desbloqueo del dispositivo para evitar que se utilicen fotografías de la víctima durante el reconocimiento facial. Adicionalmente, en junio de 2012 Google presentó una patente [11] que finalmente se aceptó en junio de 2013. La patente propone una mejora en la seguridad del control de acceso facial añadiendo la posibilidad de usar gestos en la detección facial. El usuario deberá establecer un gesto determinado para poder desbloquear el dispositivo, de esta manera el atacante además de disponer de una fotografía de la víctima tendría que simular la mueca escogida por el usuario. Algunos ejemplos son sacar la lengua, sonreír o bien mover las cejas. Con esta medida resulta más complejo llevar a cabo el ataque ya que se deberían utilizar técnicas de edición

Figura 1. Cargador móvil solar

de imagen para simular la acción del usuario propietario del dispositivo.

V. Recomendaciones para la mejora de la seguridad

A raíz de los casos del apartado IV, se derivan algunas cuestiones relativas a la seguridad y la usabilidad de los dispositivos móviles en el entorno *Android* en las cuales queremos incidir particularmente. En esta sección propondremos algunas acciones que permitan mejorar la seguridad de los dispositivos.

El caso IV-A señala la importancia de proteger adecuadamente el sistema de almacenamiento del dispositivo, así como valorar correctamente cuáles son las acciones que puede realizar una aplicación.

V-A. Almacenamiento externo: La primera recomendación hace referencia al acceso al almacenamiento externo ya que este no se rige por los sistemas descritos en la subsección III-C. A menos que haya un cambio sustancial en el sistema de acceso a datos de la memoria externa, creemos que la información sensible no debería ser almacenada en medios compartidos. Eso implica necesariamente una revisión del sistema operativo en la gestión de archivos que a su vez, debería incidir en el diseño de las aplicaciones que utilicen dispositivos de almacenamiento externo. Teniendo en cuenta que los cambios en el *SO* no son decisión del usuario final, aunque la comunidad de desarrolladores puede incentivarlos, proponemos otra solución que sí está al alcance del usuario, el cifrado del dispositivo y la *SD Card*.

V-B. Cifrado del dispositivo y la SD Card: Si optáramos por cifrar el dispositivo obtendríamos una solución parcial a la vulnerabilidad descrita anteriormente. Durante la utilización del *smartphone* o la tableta cualquier aplicación maligna continuaría siendo una amenaza ya que los ficheros no se encontrarían protegidos. Por otra parte, en caso de pérdida o robo, nuestra información permanecería inaccesible. Sin embargo, si eligiéramos cifrar la *SD Card*, conseguiríamos una protección completa frente la vulnerabilidad presentada ya que durante la utilización del *smartphone* o la tableta, una aplicación maligna que intentara acceder a nuestros datos necesitaría la contraseña de descifrado. Aun siendo una propuesta válida para mitigar la amenaza, ésta conlleva un empeoramiento en el nivel de usabilidad debido a que se produce una ralentización en el tratamiento de ficheros, así como se requieren más autorizaciones por parte del usuario durante la realización de acciones.

V-C. Antivirus: La tercera solución que proponemos es proteger el dispositivo usando un antivirus. Actualmente en la plataforma *Android* hay una gran variedad de soluciones que realizan escaneo de *malware*, protección de navegación web, así como escaneo de aplicaciones y contenidos en la *SD Card*, entre otras cosas. Aun así, un estudio realizado por V. Rastogi et al. [12] demuestra que este tipo de *software* debe mejorar ya que es susceptible a ataques de transformación por parte del *malware* y virus residentes en los dispositivos. Por otra parte, la eficacia de la aplicación antivirus se ve muy reducida en caso de que esta no posea privilegios de usuario administrador. En estas circunstancias, la aplicación no conseguiría monitorizar las actividades de otras aplicaciones debido a la falta de privilegios, y a su vez, tampoco podría monitorizar las actividades consideradas como más peligrosas. Así pues, en caso de utilizar esta opción, se recomienda *rootear* el dispositivo previamente.

V-D. Sistema de permisos: Por último, la solución más simple y plausible sería adoptar cambios en la gestión del sistema de permisos tal y como se describe en la sección III-D, empezando por una mejor concienciación por parte del usuario sobre qué tipo de permisos requiere una aplicación y como ésta puede manipular sus datos personales. Seguido de una modificación del sistema que permita aceptar o rechazar un subgrupo de permisos de manera que el usuario no se deba aceptar todo el conjunto para así poder utilizar la aplicación deseada. Éste es un problema muy común y una de las principales fuentes de amenazas debido a la inexperiencia o poco conocimiento técnico de las personas que utilizan dispositivos móviles con plataforma *Android*.

El caso IV-B demuestra la importancia de los mecanismos de control de los dispositivos.

V-E. Depuración USB: Sólo deberían tener activada los desarrolladores, y utilizarla con suma cautela. A través de la

conexión *USB* con el *ADB* se pueden realizar muchas acciones que comprometen la seguridad y privacidad de los datos del dispositivo. Hacer una copia completa del dispositivo, eliminar datos o bien instalar aplicaciones *malware* son algunos de los ejemplos más comunes. Hasta la versión 4.2.2 la amenaza solo podría ser contrarrestada en caso de tener la depuración *USB* deshabilitada. A partir de la versión 4.2.2 hasta la 4.3 el acceso al dispositivo está sujeto a mayor control ya que además se debe confirmar la conexión mediante un diálogo basado en el algoritmo *RSA*, para ello se precisa desbloquear el control de acceso del dispositivo (III-C). En caso de que el usuario no haya establecido ningún mecanismo de protección de acceso, el dispositivo sufriría la misma vulnerabilidad que en las versiones anteriores de la plataforma ya que cualquier individuo con acceso físico al dispositivo podría aceptar el diálogo *RSA* y confirmar la conexión. Cabe destacar que no es conveniente utilizar la opción *"Permitir siempre en este ordenador"* en el diálogo *RSA* ya que eso inhabilitaría la protección adicional que nos proporcionan las versiones más recientes del sistema.

V-F. Conexión a otros dispositivos: Los dispositivos tienen la capacidad de interconectarse entre ellos usando tecnologías diversas como *Bluetooth*, *USB*, *Wi-Fi*, *NFC*... Así pues nuestra recomendación es tener activadas estas tecnologías únicamente cuando sea necesario, con ello conseguiremos tener nuestro dispositivo más protegido frente amenazas del entorno. De lo contrario, se podría dar el caso donde terceros accedieran a nuestros datos o podríamos sufrir infecciones de virus, *malware*, rootkits, etc. Además, también debemos ser conscientes sobre la importancia de conectar nuestros dispositivos a otros dispositivos de confianza, de lo contrario nos podríamos exponer a amenazas como las que hemos tratado en el caso IV-A o IV-B.

V-G. Interfaz de login: Todos los usuarios deberían tenerla activada. Aunque las contraseñas o combinaciones de dígitos podrían verse expuestas a ataques de fuerza bruta o ataques de diccionario, de momento se consideran las opciones más robustas en el mecanismo de control de acceso. Los métodos usando parámetros biomédicos no son lo suficientemente eficaces como para adoptarlos como métodos de control de acceso. De hecho, el mismo dispositivo ya lo advierte en el momento de elegir el mecanismo y tal como se indica en el apartado IV-C se ha conseguido romper el control de acceso.

VI. Conclusiones y líneas futuras

Del análisis realizado y expuesto en este artículo se constata que la plataforma *Android* aún teniendo versiones comerciales estables y bastante seguras, debe mejorar e incrementar los métodos de seguridad utilizados en los dispositivos. De lo contrario, seguiremos encontrando vulnerabilidades (algunas de ellas graves) que afecten a un gran número de usuarios. Además, las vulnerabilidades detectadas, o bien la falta de implementación de algunas técnicas de seguridad, pueden provocar que la plataforma *Android* no sea considerada como una opción empresarial en beneficio de otras plataformas existentes como *Windows Phone*, *BlackBerry* o *iOS*.

Cabe destacar también la necesidad de compromiso entre seguridad y usabilidad, tan importante es un sistema seguro como suficientemente práctico y amigable para el usuario. En esta dirección el sistema *Android* debería asumir la necesidad de realizar algunos cambios estructurales que se han comentado anteriormente en la subsección V como por ejemplo el sistema de permisos de las aplicaciones.

Se han identificado algunas líneas futuras para profundizar el estudio realizado en este trabajo. Usualmente, las vulnerabilidades de seguridad se corrigen por parte de los desarrolladores cuando estas son detectadas. Aun así, es necesario un método general para poder identificarlas y evitar que los programadores de aplicaciones las introduzcan inadvertidamente. Por otro lado, la protección contra alguno de estos ataques (como por ejemplo, la identificación de la cara del usuario) depende directamente de la potencia de procesado del hardware.

Finalmente, por falta de espacio no se han incluido en el estudio otras vulnerabilidades identificadas, como la posibilidad de modificar el sistema operativo sin el consentimiento del usuario, aplicaciones capaces de eludir el sistema POSIX de permisos o aplicaciones que pueden ser modificadas por un atacante incluso después de su firmado digital.

Agradecimientos

Este trabajo se ha financiado en parte por los proyectos TAMESIS (TEC2011-22746) y ARES (CSD2007-00004).

Referencias

[1] J. Fingas, "Android climbed to 79 percent of smartphone market share in 2013, but its growth has slowed," January 2014. [Online]. Available: http://www.engadget.com/2014/01/29/strategy-analytics-2013-smartphone-share/

[2] A. Shabtai, Y. Fledel, U. Kanonov, Y. Elovici, and S. Dolev, "Google Android: A State-of-the-Art Review of Security Mechanisms," *ArXiv e-prints*, Dec. 2009.

[3] E. Chin, A. P. Felt, V. Sekar, and D. Wagner, "Measuring user confidence in smartphone security and privacy," *Proceedings of the Eighth Symposium on Usable Privacy and Security - SOUPS '12*, no. 1, p. 1, 2012. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2335356.2335358

[4] Google, "Notes on the implementation of encryption in android 3.0," 2014. [Online]. Available: https://source.android.com/devices/tech/encryption/android_crypto_implementation.html

[5] ——, "Android 4.3 apis," 2014. [Online]. Available: http://developer.android.com/about/versions/android-4.3.html

[6] ——, "Platform versions," 2014. [Online]. Available: http://developer.android.com/about/dashboards/index.html

[7] ——, "Detecting pirated applications," Febrero 2014. [Online]. Available: http://www.google.com/patents/EP2693356A2?cl=en

[8] B. Bosschert, "Steal whatsapp database (poc)," Marzo 2014. [Online]. Available: http://bas.bosschert.nl/steal-whatsapp-database/

[9] ——, "Steal whatsapp update," Marzo 2014. [Online]. Available: http://bas.bosschert.nl/steal-whatsapp-update/

[10] US-CERT/NIST, "Vulnerability summary for cve-2013-6271," Diciembre 2013. [Online]. Available: http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2013-6271

[11] Google, "Facial recognition," Junio 2013, pN 8457367. [Online]. Available: http://patft1.uspto.gov/

[12] V. Rastogi, Y. Chen, and X. Jiang, "Evaluating Android Anti-malware against Transformation Attacks," *NORTHWESTERN University*, no. March, 2013. [Online]. Available: https://www.eecs.northwestern.edu/docs/techreports/2013_TR/NU-EECS-13-01.pdf

Identificación de la Fuente en Vídeos de Dispositivos Móviles

David Manuel Arenas González, Jocelin Rosales Corripio, Ana Lucila Sandoval Orozco, Hiram Jafet Romo Torres, Luis Javier García Villalba

Grupo de Análisis, Seguridad y Sistemas (GASS), Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid
Email: {darenas, jocerosa, asandoval, javiergv}@fdi.ucm.es, hromot@hotmail.com

***Resumen*—La realización de vídeos con dispositivos móviles se ha convertido en una actividad común dado su alto grado de utilización y el gran número de usuarios. Además, la portabilidad de este tipo de dispositivos hace que estén a mano de los usuarios gran cantidad de tiempo facilitando que se utilicen para generar vídeos en una gran diversidad de situaciones. Por tanto, estos vídeos pueden ser utilizados como evidencias en procesos judiciales. Todo lo anterior hace necesario contar con técnicas de análisis forense enfocadas en vídeos de dispositivos móviles dada las características peculiares de sus cámaras. En este trabajo se estudia la identificación de la fuente de adquisición de los vídeos de dispositivos móviles y se presenta una técnica basada en la extracción del ruido del sensor y la transformada wavelet de los fotogramas extraídos del vídeo. Estos fotogramas son extraídos mediante un algoritmo que tiene en cuenta la naturaleza de los mismos, mejorando la selección de los fotogramas a analizar. Finalmente se presentan experimentos con vídeos de dispositivos móviles para evaluar la validez de las técnicas utilizadas.**

***Palabras clave*—Análisis forense de vídeos, fuente de adquisición de vídeos, patrón de ruido del sensor, PRNU. *(Video forensics analysis, video source adquisition, sensor pattern noise, PRNU)*.**

I. Introducción

Si las imágenes capturadas por dispositivos electrónicos son consideradas parte de la verdad como hechos reales, en pocos minutos, un vídeo puede comunicar una enorme cantidad de información. Según el medidor de tráfico "*Alexa, The Web Information Company*" [1], Youtube es actualmente el tercer sitio con más visitas del mundo, lo cual nos deja un claro indicio de la popularidad de la que gozan los vídeos entre los diferentes medios en los que puede desplegarse. Existe una amplia gama de dispositivos móviles que pueden reproducirlo y/o grabarlo, como por ejemplo: teléfonos móviles, tablets, vídeoconsolas portátiles y cámaras digitales o de vídeo. En cuanto a los dispositivos móviles, *Gartner Inc.* [2], afirma que las ventas de teléfonos inteligentes creció un 36 % en el cuarto trimestre del 2013. Asimismo, este tipo de dispositivos representó el 57.6 % de las ventas globales de teléfonos móviles en el cuarto trimestre de 2013, frente al 44 % del año anterior. Al igual que las cámaras digitales han desplazado en términos de uso a las cámaras tradicionales de película, actualmente, los dispositivos móviles equipados con cámaras, tienen un papel importante poniendo fin al rápido crecimiento de las cámaras digitales. En los dispositivos móviles, se ha visto una gran competencia entre fabricantes que se esfuerzan en integrar una videocámara de alta definición al alcance del usuario. Como consecuencia de este fenómeno y de la gran cantidad de tiempo que una persona pasa junto a un teléfono inteligente, este se ha convertido en el primer dispositivo de grabación de vídeos para muchos usuarios en la sociedad actual.

Debido al frecuente uso de los dispositivos móviles, en ciertos casos existen restricciones legales sobre el uso de este dispositivo, así como también de su uso en distintos lugares, tales como: colegios, universidades, oficinas de gobierno, empresas, etc. Actualmente los vídeos se exhiben con mayor frecuencia, ya sea directa o indirectamente en procesos judiciales como pruebas o evidencias para la aplicación de la ley [3]. Por tanto, dada la importancia de los vídeos en estas situaciones, el análisis forense cobra especial relevancia. Dentro de las distintas ramas del análisis forense, destaca la que nos permite identificar la fuente de adquisición, en este caso de la videocámara que generó el vídeo. En este trabajo se presentan técnicas de análisis forense para la identificación de la fuente de adquisición de vídeos, centrándonos especialmente en los vídeos generados por dispositivos móviles.

Este trabajo está estructurado en 6 secciones, siendo la primera de ellas la presente introducción. En la sección 2 se presentan brevemente las diferencias entre el pipeline en la creación de una imagen y un vídeo. La sección 3 realiza un estado del arte del análisis forense para imágenes y vídeos generados por dispositivos móviles. En la sección 4 se presenta la técnica propuesta. Los experimentos realizados y sus resultados son presentados en la sección 5. Por último en la sección 6 se presentan las conclusiones obtenidas de este trabajo.

II. Pipeline de una Videocámara

Antes de mencionar alguna de las técnicas existentes para la identificación de la fuente, es importante comprender cuál es el procedimiento realizado para generar un vídeo. Este proceso es similar en la generación de una imagen y de un vídeo, salvo que en un vídeo finalmente existe un último paso que consiste

en codificar los fotogramas resultantes para la creación de un archivo único final de vídeo. Esta codificación tiene como objetivo transformar todos los fotogramas capturados en una secuencia de ellos a lo largo de un tiempo. También se busca conseguir un tamaño lo más óptimo posible del archivo final, ya que en un vídeo existen fotogramas capturados que son redundantes entre sí. Es decir, en ocasiones entre un fotograma y otro, se puede compartir características de la escena que facilitan el poder optimizar el tamaño del vídeo final sin perder contenido visual. Por ejemplo, para la codificación MPEG, existe una estructura llamada GOP (*Group of Pictures*) que especifica el orden en el que las imágenes son ordenadas y soluciona el problema de redundancia en la codificación. Para la codificación habitualmente se utilizan los códecs: MPEG-x o H.26x para cámaras digitales y 3GP para teléfonos móviles, este último tiene la capacidad de ser compatible con los códecs: MPEG-4, H.263 o H.264 [4].

III. Técnicas de Identificación de la Fuente

La mayor parte de las investigaciones realizadas sobre la identificación de la fuente se han realizado para imágenes fotográficas estáticas. La mayoría de las técnicas que se pueden aplicar a una imagen se pueden emplear con los diferentes fotogramas de un vídeo [5].

En [6] se realiza una comparación detallada de los principales grupos de técnicas de identificación de fuente de adquisición. Estas se dividen en cinco grupos y están basadas en: metadatos, características de la imagen, defectos de la matriz CFA e interpolación cromática, imperfecciones del sensor y las transformadas wavelet.

El área que está basada en metadatos, es la más sencilla de analizar, aunque depende en gran medida de los datos que inserta el fabricante. Asimismo la agregación de metadatos a la imagen no es obligatoria. En [7], [8], [9] y [10] se utilizan los metadatos con fines de clasificación de imágenes digitales.

En [11] se trata el tema de la identificación de la fuente utilizando las características de la misma. Se contemplan tres tipos de características: características de color, características de calidad y características de la imagen en el dominio de la frecuencia. La clasificación de las imágenes es realizada por una Maquina de Soporte Vectorial (SVM). El resultado obtenido para una clasificación de cuatro cámaras de dos fabricantes distintos con contenidos similares en la imágenes fue del 100 %, mientras que para la clasificación de imágenes con contenidos distintos entre sí fue 93.05 %. Un último experimento se realizó con un conjunto de 8 cámaras que alcanzo una precisión del 95,46 %.

En [12] se utiliza una técnica que se basa en los algoritmos propietarios de interpolación cromática, los cuales dejan correlaciones a través de los planos de bits adyacentes de una imagen. Estos pueden ser representados mediante un conjunto de 108 métricas de similitud binarias y 10 métricas de calidad de la imagen (IQM). Con un clasificador KNN se realizan experimentos utilizando 9 cámaras de teléfonos móviles y 200 fotos de cada una. Para el entrenamiento se utilizaron 100 fotos de cada cámara y las 100 restantes para las pruebas. Se obtuvo un rendimiento promedio del 93.4 % de 16 experimentos que se realizaron. Hay diversos grupos de investigación que han aportado en esta área, en donde se presentan buenos resultados, por ejemplo en [13], [14] y [15].

Dentro de los métodos existentes que se basan en las imperfecciones del sensor, hay dos grandes ramas de las cuales se pueden trabajar: defectos del pixel o patrón de ruido del sensor. En [16] se demostró que los sensores de las cámaras generan un patrón de ruido (*Sensor Pattern Noise*) que podría ser utilizado como método único de identificación.

En [17] se demostró que el ruido del sensor extraído de las imágenes podían ser severamente contaminado por los detalles de las escenas concretas. Para lidiar con ese problema, se propuso un nuevo enfoque para la atenuar la influencia del detalle de las escenas en el ruido del sensor mejorando la tasa de acierto. En los experimentos se tomaron 9 cámaras y 320 fotos de cada una, variando las escenas al aire libre e interiores. En [18], [19] y [20] se presentan otros métodos de identificación de fuente basados en las imperfecciones del sensor.

Por último, en el área de las transformadas wavelets existen diversos enfoques. Por ejemplo en [21] se propone una nueva técnica de identificación basada en las características de probabilidad condicional. Este tipo de características fueron propuestas inicialmente para propósitos de estegoanálisis en [22]. Se obtuvieron unos resultados del 98.6 %, 97.8 % y 92.5 % de acierto en la clasificación de 2, 3 y 4 iPhones respectivamente con un recorte de imagen de 800x600.

En [23] se determina que el uso del patrón de ruido del sensor conjuntamente con la transformada wavelet es un método efectivo para la identificación de fuente, alcanzando una tasa de éxito promedio del 87.21 %.

En el caso del desarrollo de técnicas para la identificación de fuente de vídeo, existen pocas referencias al respecto. Algunas se basan directamente en la secuencia de codificación y otras en la extracción de frames aplicando algún método de clasificación para imágenes fijas.

En [24], se propone un algoritmo en base a la información del vector de movimiento en el flujo codificado. En los experimentos realizados se utilizaron 100 secuencias de vídeo (20 de ellas procedentes de VQEG (*Video Quality Experts Group*)[25] y 80 de DVDs). Todos los vídeos fueron codificados por diferentes aplicaciones de edición de vídeo conocidos. Mediante un experimento se obtuvo un 74.63 % de precisión en la identificación del software que se utilizó en la codificación.

En [26] propone un método de identificación utilizando los fotogramas extraídos de vídeos. Las características de probabilidad condicional se extraen directamente de los fotogramas del vídeo. En las pruebas realizadas se utilizaron 4 modelos diferentes de cámaras y un clasificador SVM, obteniendo, en un primer experimento aplicado en el dominio del espacio con los valores de luminancia, un 82.6 % de precisión. En un segundo experimento usando el mismo conjunto de vídeos, tomando el valor de luminancia, el promedio de clasificación fue de 100 %. En un tercer experimento en donde se utilizaron

un conjunto de vídeos con mayores cambios en las escenas se obtuvo un 97.2 % de acierto.

IV. Descripción de la Técnica

Al completar la generación de un vídeo es posible que se introduzcan, en cada uno de sus fotogramas, algunos defectos que se vean reflejados como ruido, llamados comúnmente "huellas". Estas "huellas" se pueden utilizar para detectar la fuente de adquisición del vídeo. A continuación, se describe el método de extracción de fotogramas del vídeo y se referencia el algoritmo que permite extraer el patrón de ruido del sensor.

Los fotogramas seleccionados para la clasificación son obtenidos mediante el algoritmo 1.

Algoritmo 1: Algoritmo de extracción de fotogramas

Input: vídeo: Vídeo a procesar
nFotogramas: Número de fotogramas deseados
$umbral_I$: Umbral inicial
inc_U: Incremento del umbral
Result: fotogramas: Vector de fotogramas

1. histogramas ← extraerHistogramas(vídeo);
2. umbral ← estimarUmbral(histogramas, nFotogramas, $umbral_I$, inc_U);
3. fotogramas ← extraerFotogramas(vídeo, umbral, histogramas);

El algoritmo calcula y compara los fotogramas contenidos en un vídeo. Los fotogramas que presenten un cambio de escena significativo serán utilizados para la clasificación e identificación. Esto se debe a que el ruido del sensor extraído de una imagen puede estar severamente contaminada por los detalles de la escena [17], además de que los datos de un vídeo contienen redundancia temporal, espacial y espectral. En el algoritmo son necesarios 4 parámetros para su funcionamiento:

- Video del cual serán extraídos los fotogramas.
- Numero de fotogramas deseados a extraer.
- Umbral inicial que será la referencia para determinar cuando existe un cambio de escena.
- Valor del incremento para el umbral que se realizará en cada iteración.

El primer paso consiste en extraer el histograma (frecuencia de los valores de color) de los fotogramas, y calcular mediante la correlación de cada par de fotogramas contiguos la similitud existente. La correlación se calcula con la ecuación (1).

$$correlacion(H_1, H_2) = \frac{\sum_i {H'}_1(i){H'}_2(i)}{\sqrt{\sum_i {H'}_1(i)^2 {H'}_2(i)^2}} \quad (1)$$

donde,

$${H'}_k(i) = H_k(i) - \frac{1}{N}(\sum_j H_k(j))$$

siendo N el número de niveles de intensidad para cada canal de color RGB.

Existen diversos métodos para calcular la diferencia de histogramas de color de dos dimensiones. En este trabajo se optó por el cálculo de la correlación, ya que es un vector aleatorio (variable aleatoria multidimensional) y además, los resultados obtenidos con el coeficiente de correlación son mejores que otras medidas [27]. Se puede mencionar por ejemplo la distribución de probabilidad continua (chi-cuadrado) o la intersección o distancia de Bhattacharyya.

El primer fotograma del vídeo se toma como parte del conjunto de fotogramas elegidos. Se realiza la comparación tomando el primer y el segundo fotograma, si no hay una diferencia significativa entre ellos en base al umbral, se toma el siguiente fotograma y se realiza una nueva comparación con el primero, esto se realiza hasta que el resultado de la correlación sea menor al umbral, para tomar en cuenta al fotograma para la clasificación e identificación. Si al final la cantidad de cambios de escena en base al umbral, es menor a la cantidad necesaria, se repite el proceso de comparación incrementando el umbral, hasta que la cantidad de cambios de escena sea mayor o igual a los deseados.

Para poder determinar el umbral inicial, se realizaron varios experimentos sobre los vídeos, y se halló que mediante la comparación de los histogramas de un vídeo, la correlación promedio más baja fue de -0.27, presentando al menos 1 o 2 cambios de escena, definiendo así el umbral inicial. Para el valor del incremento se experimentó con diferentes valores, tales como: 0.1, 0.01, 0.001, 0.0001 y 0.0001. El valor de 0.001 fue el elegido, ya que demostró ser un valor ideal para llegar a el número de fotogramas deseados en un menor tiempo y con más exactitud. Estos incrementos se realizan porque, si el umbral se encuentra más cercano al valor máximo de correlación directa, es decir al valor de 1, se pueden encontrar más cambios de escena, y así extraer la cantidad de fotogramas definidos por el usuario para la clasificación e identificación.

Mediante el análisis de los trabajos de la literatura, se llegó a la conclusión de que el patrón de ruido del sensor y la transformada wavelet, ayudan a definir una huella, siendo métodos efectivos para la identificación de fuente. Este articulo extiende el uso del patrón de ruido del sensor y la transformada wavelet de [23]. La técnica está enfocada en representar las huellas en vectores de características.

El esquema presentado en la Figura 1 muestra el diagrama funcional de la técnica.

La obtención del patrón de ruido del sensor de las imágenes, se basa en el método descrito en [23].

El siguiente paso es obtener las características que caracterizan el ruido del sensor para fines de la clasificación. Un total de 81 características son obtenidas utilizando el algoritmo de extracción de características descrito en [23].

V. Experimentos y Resultados

Para probar la efectividad del técnica propuesta, se capturaron vídeos sin ninguna consideración en las características temporales o espaciales, debido a que deben representar casos reales. Como actualmente los dispositivos móviles presentan grandes mejoras en la calidad del vídeo, se consideró usar

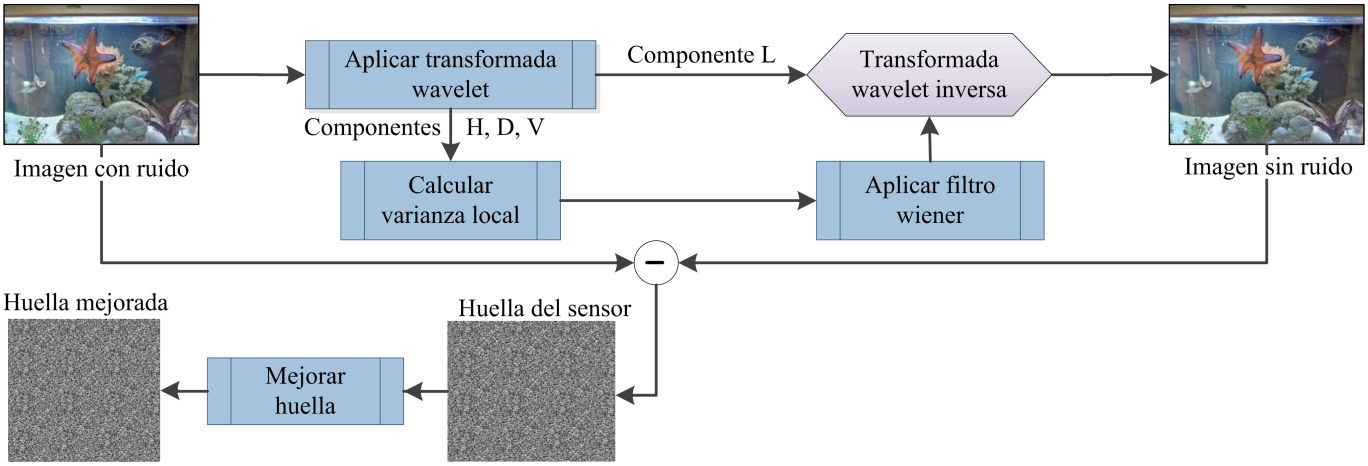


Figura 1. Esquema funcional de extracción de patrón de ruido del sensor.

vídeos con calidad de 1080p (vídeos de alta definición), es decir con una resolución de 1920x1080 pixeles.

La Tabla I muestra las especificaciones básicas y los modelos de dispositivos móviles considerados para los experimentos.

Tabla I
CONFIGURACIÓN DE LAS CÁMARAS DE DISPOSITIVOS MÓVILES

Marca (Modelo)	FPS	Formato	Códec	Condiciones de Captura
Apple iPhone 5 (M1)	24	mov	H.264	Resolución: 1080p Tipo de escena: Cualquiera Orientación: Vertical Flash: Deshabilitado Luz: Natural Balance de blancos: Autom Zoom digital: 0 Tiempo de vídeo: 2 min
Nokia 808 Pureview (M2)	30	mp4	MPEG-4	
Samsung Galaxy S4 (M3)	30	mp4	MPEG-4	
Wiko Cink Slim (M4)	12	3gp	MPEG-4	
Zopo ZP-980 (M5)	15	3gp	MPEG-4	

En la Tabla II se resumen las condiciones experimentales usadas en la evaluación de la técnica propuesta.

La clasificación se ha realizado utilizando el método de aprendizaje supervisado SVM con kernel RBF, ya que es uno de los más utilizados para este tipo de clasificaciones [6]. Los parámetros utilizados en el clasificador SVM son los mismos que los empleados en [23]. Se utilizó el paquete LibSVM [28] que permite la clasificación de múltiples clases.

Con la finalidad de mostrar a grandes rasgos tiempos de ejecución para la obtención del conjunto de características, para una imagen con un recorte de 1024x768, en un procesador Intel Core i7 de 1.6GHz con 8Gb de RAM, se consumen aproximadamente 2 segundos. Para 500 imágenes con el recorte anterior y utilizando la misma máquina, en las fases de entrenamiento y clasificación de la SVM se emplearon aproximadamente unos 650 segundos y 1 segundo respectivamente.

Tabla II
PARÁMETROS DE LOS EXPERIMENTOS

Parámetro	Valor
Numero de vídeos para entrenamiento por cámara	5
Numero de vídeos para pruebas por cámara	5
Método de extracción	Histograma
Umbral inicial	-0.27
Incremento umbral	0.001
Numero de fotogramas deseados por vídeo	100

V-A. *Influencia de la Resolución en la Tasa de Acierto*

Para analizar la influencia que tiene tamaños de recortes de los fotogramas en la tasa de acierto, se realizó la identificación de la fuente de los 5 dispositivos móviles de la Tabla I con cada una de las siguientes resoluciones de los fotogramas:

- *Resoluciones estándar*: 128x128, 320x240, 640x480, 800x600, 1024x768.
- *Resolución recomendada*: 1024x1024
- *Resolución real del fotograma*: 1920x1080.

Los parámetros utilizados para la extracción de las características definido en [23] son: Daubechies 8 wavelet, recorte del fotograma centrado y estimación de varianza adaptativa.

En la Tabla III se muestra el porcentaje de acierto medio en la identificación de la fuente de cada dispositivo con respecto a los distintos tamaños de recortes de los fotogramas. Por porcentaje de acierto se entiende al porcentaje de fotogramas de los 5 vídeos de cada dispositivo móvil clasificados correctamente. Cada vídeo obtuvo un porcentaje de acierto en la identificación de la fuente que se muestra en la tabla III.

En la mayoría de los casos, los porcentajes de acierto por dispositivo, aumentan cuanto más grande sea el recorte de los fotogramas (esto se da para todos los casos si se tiene en cuenta la tasa de acierto promedio). Obteniendo para la mayor resolución la mayor tasa de acierto promedio, un 85.56 %. En todos los experimentos realizados se supera la tasa por vídeo individual del 50 %. Esto indica que en todos los casos, para

Tabla III
TASA DE ACIERTO × RESOLUCIÓN DE FOTOGRAMAS.

Resolución	Dispositivos					% Acierto
	M1	M2	M3	M4	M5	Medio
128x128	65.80 %	79.20 %	66.60 %	75.00 %	64.00 %	70.12 %
320x240	73.80 %	88.20 %	78.00 %	78.80 %	63.40 %	76.44 %
640x480	79.60 %	95.60 %	85.00 %	85.60 %	66.20 %	82.40 %
800x600	81.80 %	96.80 %	84.80 %	86.00 %	68.80 %	83.64 %
1024x768	80.80 %	97.40 %	88.80 %	85.40 %	75.40 %	85.56 %
1024x1024	**81.20 %**	**97.20 %**	**92.20 %**	**88.00 %**	**78.40 %**	**87.40 %**
1920x1080	87.40 %	98.80 %	93.00 %	89.80 %	82.60 %	90.32 %

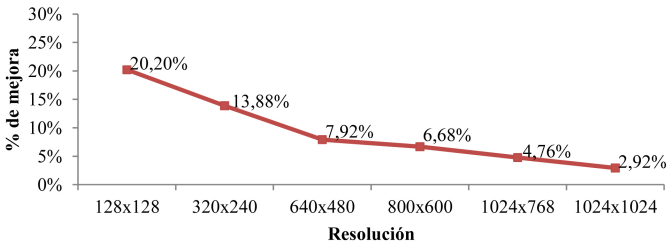


Figura 2. Porcentaje de mejora de la tasa de acierto × por resolución.

todos los fotogramas de un vídeo concreto de un dispositivo concreto, al menos el 50 % de los fotogramas son identificados correctamente. Finalmente la identificación de la fuente de un vídeo debe responder a la pregunta concreta de a qué fuente de adquisición pertenece ese vídeo. Como criterio lógico, puede estimarse que el vídeo pertenece a la fuente con el mayor número de fotogramas clasificados con respecto a las otras fuentes (mayor porcentaje de acierto con respecto a las otras fuentes). Se podría dar el caso en el que varias fuentes tengan exactamente el mismo número de fotogramas clasificados y a su vez sean el mayor número con respecto a las otras fuentes. En este caso, poco habitual, se diría que la fuente del vídeo no puede ser identificada con determinación y estaría entre la duda de esas distintas fuentes.

Los resultados obtenidos no dejan lugar a dudas sobre la identificación de la fuente de adquisición del vídeo teniendo en cuenta el criterio definido anteriormente, ya que en todos los casos el acierto superó el 50 %. Asimismo puede verse que las tasas de acierto en muchos casos son mucho mayores (llegando en ocasiones hasta el 100 %). Por tanto, según este experimento, tomando el criterio antes definido y teniendo en cuenta el vídeo como entidad unitaria (es decir un vídeo se clasifica bien o no), se puede concluir que esta técnica identifica la fuente de un vídeo con un 100 % de acierto.

Como se puede observar en la Tabla III, utilizando la imagen completa existe una mayor tasa de acierto promedio en la identificación de fuente, aunque el incremento es pequeño. Sin embargo, la influencia que tiene la resolución en la tasa de aciertos de la identificación de la fuente que adquirió un vídeo se refleja en la Figura 2, donde se muestra que la mejora en la tasa de acierto para la tamaño de recorte de 1920x1080 es del 2.92 % con respecto a un tamaño de recorte de 1024x1024. Por tanto, a partir de un cierto tamaño de recorte el incremento de la tasa de acierto es pequeña, e incluso en algunos casos este puede disminuir un poco. También hay que tener en cuenta que a mayor tamaño de recorte mayor tiempo de ejecución del algoritmo de extracción de características.

V-B. Influencia de los Parámetros de Ejecución en la Tasa de Acierto

Para analizar cómo afecta el uso de los distintos parámetros del algoritmo propuesto, en [23] en la identificación de la fuente de vídeos, se realizó un conjunto de experimentos utilizando el mismo tamaño de recorte centrado del fotograma (640x480) para identificar la fuente de los 5 dispositivos móviles de la Tabla I. En la Tabla IV se muestra un resumen de los experimentos realizados y los parámetros de configuración del algoritmo de extracción de las características utilizados en cada uno de ellos.

Tabla IV
TASA DE ACIERTO UTILIZANDO DIFERENTES CONFIGURACIONES

Configuración	Varianza	Aplicar Zero Meaning	% de Acierto
1	Adaptativa	No	82.4 %
2	Adaptativa	Sí	82.32 %
3	No adaptativa	Sí	81.56 %
4	No adaptativa	No	82.96 %

La diferencia entre la mejor y la peor tasa de acierto mostrada en la Tabla IV es del 1.4 %. Este resultado indica que los parámetros de configuración del algoritmo de extracción de características no influye significativamente en el porcentaje de acierto de identificación de la fuente de vídeos. Puede verse que la tasa óptima de acierto se consigue con los parámetros de estimación de la varianza no adaptativa y sin utilizar el filtro zero-meaning. Asimismo los peores resultados se obtuvieron con los parámetros de estimación de varianza no adaptativa y el uso del filtro zero-meaning. Dado el estrecho margen entre las tasas de acierto que hay entre las diferentes configuraciones y los resultados de la configuración óptima y las restantes, las conclusiones no pueden extrapolarse de forma categórica para cualquier experimento, aunque éstas deben de tenerse en cuenta para futuros experimentos y aplicación de la técnica.

VI. CONCLUSIONES

Una vez presentada la técnica y realizados los experimentos variando los distintos parámetros, se llega a la conclusión general de que esta técnica obtiene buenos resultados y es válida para la identificación de la fuente en vídeos de dispositivos móviles. La aplicación a escenarios reales de esta técnica la consideramos realista y viable, siempre que los videos a clasificar pertenezcan a un conjunto cerrado y conocido de dispositivos móviles.

El algoritmo de extracción de fotogramas presentado tiene en cuenta la naturaleza de un vídeo y sus fotogramas, optimizando la extracción de los fotogramas claves. Es decir, extrae los fotogramas teniendo en cuenta que si los fotogramas obtenidos tienen mayor variación de escena entre ellos (buscando los cambios de escena), el proceso de clasificación

obtendrá mejores resultados. Sin embargo, para la clasificación utilizando SVM se necesita un número determinado de fotogramas para el entrenamiento, y esto el algoritmo también lo tiene en cuenta ya que se puede dar el caso en el que el vídeo cambie poco de escena y tenga que obtener los fotogramas más distantes en escenas entre los que hay. Una vez obtenidos los fotogramas nos basamos en la extracción de características que se obtienen del patrón de ruido del sensor y la transformada wavelet especificado en [23].

Los resultados promedios de clasificación varían dependiendo de los parámetros utilizados. Teniendo en cuenta un tamaño de recorte centrado, se concluye que a mayor tamaño de recorte, mejores son los resultados. Asimismo, se ha evaluado como afecta el uso de los distintos parámetros de configuración definidos en [23] en la identificación de la fuente de vídeos de dispositivos móviles usando un recorte centrado de fotograma de 640x480. En este sentido, no se han podido obtener conclusiones categóricas y extrapolables sobre el uso de los parámetros de configuración, ya que en todos los experimentos realizados la tasa de acierto están comprendida en un margen muy pequeño.

Una vez clasificados los fotogramas seleccionados, se debe responder a la pregunta de cuál es la fuente de adquisición del vídeo como entidad unitaria. Nuestro criterio ha sido que el vídeo pertenece a la fuente cuyo mayor número de fotogramas se han clasificado de ese tipo. Este criterio debe tenerse en cuenta en futuras comparaciones con otras técnicas.

Agradecimientos

El Grupo de Investigación GASS agradece la infraestructura proporcionada por el Campus de Excelencia Internacional (CEI) Campus Moncloa - Clúster de Cambio Global y Nuevas Energías (y, más concretamente, el sistema EOLO como recurso de computación de alto rendimiento HPC - High Performance Computing), infraestructura financiada por el Ministerio de Educación, Cultura y Deporte (MECD) y por el Ministerio de Economía y Competitividad (MINECO).

Referencias

[1] "Alexa Top 500 Global Sites," http://www.alexa.com/topsites, 2014.

[2] "Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013," http://www.gartner.com/newsroom/id/2665715, 2014.

[3] C. Wen and K. Yang, "Image authentication for digital image evidence," *Forensic Science Journal*, vol. 5, no. 1, pp. 1–11, September 2006.

[4] "Formatos, códecs y extensiones de archivos de audio y vídeo compatibles - Cómo - BlackBerry Q10 Smartphone - 10.1," http://docs.blackberry.com/es-es/smartphone_users/deliverables/50637/mba1344953159594.jsp, 2014.

[5] P. Bestagini, M. Fontani, S. Milani, M. Barni, A. Piva, M. Tagliasacchi, and S. Tubaro, "An Overview on Video Forensics," in *Proceedings of the 20th European Signal Processing Conference*, August 2012, pp. 1229–1233.

[6] A. Sandoval Orozco, D. Arenas González, J. Rosales Corripio, L. García Villalba, and J. Hernandez-Castro, "Techniques for Source Camera Identification," in *Proceedings of the 6th International Conference on Information Technology*, May 2013, pp. 1–9.

[7] N. L. Romero, V. G. Chornet, J. S. Cobos, A. S. Carot, F. C. Centellas, and M. C. Mendez, "Recovery of Descriptive Information in Images From Digital Libraries by Means of EXIF Metadata," *Library Hi Tech*, vol. 26, no. 2, pp. 302–315, 2008.

[8] M. Boutell and J. Luo, "Beyond Pixels: Exploiting Camera Metadata for Photo Classification," *Pattern Recognition*, vol. 38, no. 6, pp. 935–946, June 2005.

[9] J. Tesic, "Metadata Practices for Consumer Photos," *IEEE Multimedia*, vol. 12, no. 3, pp. 86–92, September 2005.

[10] M. Boutell and J. Luo, "Photo Classification by Integrating Image Content and Camera Metadata," in *Proceedings of the 17th International Conference on Pattern Recognition*, vol. 4. IEEE Computer Society, August 2004, pp. 901–904.

[11] M. J. Tsai, C. L. Lai, and J. Liu, "Camera/Mobile Phone Source Identification for Digital Forensics," in *Proceedings of the International Conference on Acoustics Speech and Signal Processing*. IEEE, April 2007, pp. II–221–II–224.

[12] O. Celiktutan, I. Avcibas, B. Sankur, N. P. Ayerden, and C. Capar, "Source Cell-Phone Identification," in *Proceedings of the IEEE 14th Signal Processing and Communications Applications*. IEEE, April 2006, pp. 1–3.

[13] Y. Long and Y. Huang, "Image Based Source Camera Identification using Demosaicking," in *Proceedings of the IEEE 8th Workshop on Multimedia Signal Processing*. IEEE, October 2006, pp. 419–424.

[14] S. Bayram, H. Sencar, N. Memon, and I. Avcibas, "Source Camera Identification Based on CFA Interpolation," in *Proceedings of the IEEE International Conference on Image Processing*, vol. 3, September 2005, pp. III–69–72.

[15] S. Bayram, H. T. Sencar, and N. Memon, "Classification of Digital Camera-Models Based on Demosaicing Artifacts," *Digital Investigation*, vol. 5, no. 1, pp. 49–59, June 2008.

[16] J. Lukas, J. Fridrich, and M. Goljan, "Digital Camera Identification from Sensor Pattern Noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, June 2006.

[17] C. Li, "Source Camera Identification Using Enhanced Sensor Pattern Noise," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 280–287, June 2010.

[18] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh, "Methods for Identification of Images Acquired with Digital Cameras," in *Proceedings on Enabling Technologies for Law Enforcement and Security*, vol. 4232. SPIE-International Society for Optical Engine, February 2001, pp. 505–512.

[19] T. Lanh, K. Chong, S. Emmanuel, and M. Kankanhalli, "A Survey on Digital Camera Image Forensic Methods," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, July 2007, pp. 16–19.

[20] F. D. O. Costa, M. Eckmann, W. J. Scheirer, and A. Rocha, "Open Set Source Camera Attribution," in *Proceedings of the Patterns and Images Proceedings of the 25th Conference on Graphics*. IEEE, August 2012, pp. 71–78.

[21] A. Wahab, A. Ho, and S. Li, "Inter-Camera Model Image Source Identification with Conditional Probability Features," in *Proceedings of the 3rd Image Electronics and Visual Computing Workshop*, November 2012.

[22] A. Wahab, J. Briffa, H. Schaathun, and A. T. S. Ho, "Conditional Probability Based Steganalysis for JPEG Steganography," in *Proceedings of the International Conference on Signal Processing Systems*, May 2009, pp. 205–209.

[23] J. Rosales Corripio, A. L. Sandoval Orozco, L. J. García Villalba, J. Hernandez-Castro, and S. J. Gibson, "Source Smartphone Identification Using Sensor Pattern Noise and Wavelet Transform," in *Proceedings of the 5th International Conference on Imaging for Crime Detection and Prevention*, December 2013, pp. 1–6.

[24] Y. Su, J. Xu, and B. Dong, "A Source Video Identification Algorithm Based on Motion Vectors," in *Proceedings of the Second International Workshop on Computer Science and Engineering*, vol. 2, October 2009, pp. 312–316.

[25] "Video Quality Experts Group (VQEG)," http://www.its.bldrdoc.gov/vqeg/vqeg-home.aspx, 2014.

[26] S. Yahaya, A. Ho, and A. Wahab, "Advanced Video Camera Identification Using Conditional Probability Features," in *Proceedings of the IET Conference on Image Processing*, July 2012, pp. 1–5.

[27] E. Bashkov and N. Shozda, "Content-Based Image Retrieval Using Color Histogram Correlation," *Graphicon Proceedings*, 2002.

[28] C. C. Chang and C. J. Lin, "LIBSVM: A Library for Support Vector Machines. Version 3.17, Abril 26, 2013," http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

Clasificación sin Supervisión de Imágenes de Dispositivos Móviles

David Manuel Arenas González, Jocelin Rosales Corripio, Ana Lucila Sandoval Orozco,
Jorge Alberto Zapata Guridi, Luis Javier García Villalba

Grupo de Análisis, Seguridad y Sistemas (GASS), Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid
Email: {darenas, jocerosa, asandoval, javiergv}@fdi.ucm.es, jorge.zapata@jazg.net

***Resumen*—Cada día el uso de imágenes de dispositivos móviles como evidencias en procesos judiciales es más habitual y común. Por ello, el análisis forense de imágenes de dispositivos móviles cobra especial importancia. En este trabajo se estudia la rama del análisis forense que se basa en la identificación de la fuente, concretamente en la agrupación o *clustering* de imágenes según la fuente de adquisición. Como diferencia con otras técnicas del estado del arte para la identificación de la fuente, en el *clustering* no se tiene un conocimiento a priori del número de imágenes ni dispositivos a identificar, ni se tienen datos de entrenamiento para una futura fase de clasificación. Es decir, se realiza un agrupamiento por clases con todas las imágenes de entrada. La propuesta se basa en la combinación de *clustering* jerárquico y plano y en el uso del patrón de ruido del sensor. Se han realizado un conjunto de experimentos que emulan situaciones similares a las que se pueden dar en la realidad para mostrar la robustez y fiabilidad de los resultados de la técnica. Los resultados obtenidos son satisfactorios en todos los experimentos realizados superando en tasa de acierto a otras propuestas descritas en el estado del arte.**

***Palabras clave*—Análisis forense de imágenes, clustering de imágenes, patrón de ruido del sensor, PRNU. *(Image forensics analysis, image clustering, sensor pattern noise, PRNU)*.**

I. Introducción

En la actualidad, el número de cámaras integradas a dispositivos móviles ha proliferado permitiendo a millones de consumidores tomar fotografías e incluso compartir de manera sencilla el contenido capturado. La industria de los dispositivos móviles ha desarrollado la tecnología necesaria para abaratar los costos y de esta manera hacerlos muy accesibles al público.

El gran número de cámaras en dispositivos móviles constituye un mayor número de evidencias presentadas ante la ley en delitos como robo de información de tarjetas de crédito, pornografía infantil, espionaje industrial, etc. Por tanto, el análisis forense de este tipo de imágenes cobra especial importancia en las investigaciones judiciales. Dentro de análisis forense de imágenes digitales existen dos grandes ramas: la identificación de la fuente de adquisición y la detección de manipulaciones malintencionadas. Este trabajo se centra en la primera rama, es decir, dada una imagen o conjunto de imágenes identificar la marca y modelo de la cámara que realizó la foto mediante la clasificación por agrupamiento o *clustering*. Asimismo, dado que las cámaras de dispositivos móviles tienen unas características propias que las hacen diferentes a la restantes, este trabajo se enfoca en las fotos de este tipo de dispositivos.

Dentro de la identificación de la fuente existen dos grandes enfoques: escenarios cerrados o escenarios abiertos. Un escenario cerrado es aquel en el cual la identificación de la fuente de la imagen se realiza sobre un conjunto de cámaras concreto y conocidas a priori. Para este enfoque normalmente se utiliza un conjunto de imágenes de cada cámara para entrenar un clasificador y posteriormente se predice la fuente de adquisición de las imágenes objeto de investigación. La técnica más utilizada para la tarea de clasificación de imágenes digitales es *Support Vector Machine* (SVM). Este trabajo se centra en la identificación de la fuente en escenarios abiertos, es decir, el analista forense no conoce a priori el conjunto de cámaras a las que pertenece la imagen a identificar su fuente. Obviamente en este tipo de clasificación, en la que no se tienen datos de cámaras a priori, el objetivo no es identificar la marca y modelo de la cámara, sino poder agrupar distintas imágenes en grupos disjuntos en los que todas sus imágenes pertenecen al mismo dispositivo. Este planteamiento es muy cercano a situaciones de la vida real, ya que en muchos casos el analista desconoce por completo el conjunto de cámaras a las que pueden pertenecer un conjunto de imágenes. Además, es prácticamente imposible tener un conjunto de imágenes para entrenar un clasificador con todas las cámaras de dispositivos móviles existentes en el mundo.

Este trabajo está estructurado en 5 secciones, siendo la primera de ellas la presente introducción. En la sección 2 se presentan brevemente los trabajos previos relacionados con las técnicas de análisis forense para la identificación de la fuente de imágenes de dispositivos móviles. En la sección 3 se presenta la técnica propuesta. Los experimentos realizados y sus resultados se presentan en la sección 4. Por último en la sección 5 se presentan las conclusiones obtenidas de este trabajo.

II. Trabajos Relacionados

La mayoría de las investigaciones realizadas sobre la identificación de la fuente de adquisición de imágenes se centran en cámaras digitales tradicionales o DSC (*Digital Still Camera*), no siendo en su mayoría estas técnicas válidas para imágenes

de dispositivos móviles. La principal razón por la que se necesitan técnicas específicas para imágenes de dispositivos móviles es que muchas de ellas se basan en la extracción de características de aspectos relacionados con el sensor. En general, los sensores de las DSC utilizan la tecnología CCD (*Charge Coupled Device*), siendo estos sensores de mayor calidad que los que utilizados en cámaras de los dispositivos móviles, los cuales se utilizan la tecnología CMOS (*Complementary Metal Oxide Semiconductor*). Dada la alta calidad de muchos de los sensores de las DSC, las técnicas forenses que utilizan las características del sensor tienen un enfoque diferente al que se utiliza para las cámaras de dispositivos móviles. Asimismo, existen otros aspectos diferenciadores entre DSC y cámaras de dispositivos móviles que se deben tener en cuenta en las distintas técnicas. Algunos de estos aspectos son el sistema de lentes y filtros o el algoritmo de interpolación utilizado. En [1] puede verse una panorámica de las distintas investigaciones realizadas.

Para cualquier tipo de clasificación de imágenes, ya sea en escenarios abiertos o cerrados, se necesita obtener ciertas características que permitan a las técnicas de clasificación realizar su tarea. Según [2] se pueden establecer cuatro grupos de técnicas para este fin: basadas en la aberración de las lentes, basadas en la interpolación de la matriz CFA, basadas en las imperfecciones del sensor y basadas en el uso de las características de la imagen. Dentro de este último grupo puede hacerse una subdivisión en las basadas en características del color (*Color Features*), características de la calidad (*Quality Features*) y estadísticas del dominio *wavelet*. Este trabajo utiliza las técnicas basadas en las imperfecciones del sensor, concretamente aquellas basadas en el patrón de ruido del sensor *Sensor Pattern Noise* (SPN) el cual es originado por las imperfecciones en el proceso de fabricación de los semiconductores o las producidas por la utilización de la cámara en el día a día.

El objetivo del análisis de *clusters* o *clustering* es agrupar una colección de objetos en clases representativas llamadas *clusters*, sin información a priori, de forma que los objetos pertenecientes a cada *cluster* guarden una mayor similitud con respecto de objetos en otros *clusters*. La agrupación de imágenes puede llevarse a cabo mediante técnicas de aprendizaje supervisadas o sin supervisión. En el primer caso es indispensable conocer información del dispositivo a priori, es decir se identifica claramente con la clasificación en escenarios cerrados en donde se requiere una fase de entrenamiento con las características extraídas de las imágenes y una segunda fase de clasificación conforme al resultado anterior. Sin embargo, en un caso real puede ser difícil contar con la cámara en cuestión o con un subconjunto de fotografías tomadas por la misma para llevar a cabo un entrenamiento, de ahí la necesidad de técnicas de aprendizaje sin supervisión, que se corresponden directamente con los escenarios abiertos. El *clustering* tradicional se caracteriza por ser una técnica de aprendizaje sin supervisión.

Para poder determinar la similitud entre objetos pertenecientes a un mismo *cluster* existen medidas de distancia como pueden ser: distancia euclideana, distancia Manhattan y distancia Chebychev, entre otras. Alternativamente, es posible usar funciones de similitud $S(X_i, X_j)$ las cuales comparan dos vectores X_i y X_j en forma simétrica, es decir, $S(X_i, X_j) = S(X_j, X_i)$. Estas funciones alcanzan sus valores más altos cuando X_i y X_j son más similares. La medida más usada en la identificación de fuente de imágenes es la correlación normalizada [3], [4], [5] definida como:

$$corr\,(X_i, X_j) = \frac{\left(X_i - \overline{X_i}\right) \odot \left(X_j - \overline{X_j}\right)}{\left\|X_i - \overline{X_i}\right\| \cdot \left\|X_j - \overline{X_j}\right\|} \qquad (1)$$

Donde $\overline{X_i}$ y $\overline{X_j}$ representan la media del vector, $X_i \odot X_j$ es el producto punto de dos vectores y $\|X_i\|$ es la norma L_2 de X_i. Dado que el patrón de ruido del sensor es una matriz bidimensional, previamente a la aplicación de las funciones del cálculo de la correlación, se realiza una transformación a vector unidimensional.

De acuerdo a la clasificación de algoritmos de *clustering* propuesta en [6] encontramos los métodos jerárquicos cuyo propósito es lograr una estructura denominada dendograma que representa la agrupación de los objetos de acuerdo a sus niveles de similitud. Esta agrupación puede realizarse de distintas formas: aglomerativa o decisiva. La agrupación aglomerativa considera inicialmente a cada objeto como una clase independiente hasta, de forma iterativa, lograr agrupar todos los objetos en una clase única. La agrupación de forma divisiva se basa en la idea de partir de una sola clase hasta lograr separar todos los objetos en clases individuales. También existen los algoritmos de particionamiento en donde iniciando de una partición, el algoritmo se encarga de mover objetos de un *cluster* a otro hasta minimizar cierto criterio de error. Dentro de esta categoría el método más famoso es el k-means, sin embargo la mayoría de estos métodos requieren conocer de antemano el número de *clusters*, por lo cual no son muy utilizados en temas de análisis forense de imágenes. Por último, existen otros algoritmos de *clustering* como: [7] que produce *clusters* por medio de grafos, [8] basado en la densidad donde los puntos dentro de un *cluster* vienen dados por cierta función de probabilidad, *clusters* basados en modelos como árboles de decisión [9] o redes neuronales [10] y *clustering* con métodos de *soft-computing* como *fuzzy clustering* [11], métodos evolucionarios de *clustering* y recocido simulado en *clustering* [12].

Existen trabajos previos sobre agrupación de imágenes por métodos sin supervisión, todos ellos consideran al SPN como el criterio más fiable para representar la huella digital de un dispositivo, es de ahí que utilizan concretamente el PRNU (*Photo Response Non-Uniformity*) como huella y la correlación normalizada como medida de similitud para lograr el agrupamiento de imágenes por dispositivo.

En [13] se utiliza una técnica de clasificación con aprendizaje no supervisado donde mediante la maximización de grafos se logra una agrupación. El *clustering* se realiza a partir de grafos no dirigidos con pesos, comenzando con una matriz de afinidad donde los pesos de conexión entre vértices es el

valor de correlación entre cada SPN, iniciando con un nodo aleatorio. En cada iteración conectan los nodos restantes y eligen los nodos más cercanos al central obteniendo una nueva matriz de afinidad en cada paso, el algoritmo se detiene cuando el número de nodos más cercanos es menor a un parámetro k. Posteriormente el grafo es particionado hasta el punto en donde la similitud en un conjunto sea máxima y mínima con respecto a otros conjuntos.

En [4] se realizan agrupamientos mediante campos markovianos aleatorios. Se propone un algoritmo de *clustering* partiendo de una matriz que contiene todas las correlaciones entre SPN de diversas cámaras. En cada iteración el algoritmo agrupa dentro de clases los SPN más similares haciendo uso de las características locales de los campos markovianos aleatorios y asigna una nueva etiqueta de clase a cada SPN maximizando una función de probabilidad. El criterio para detener el algoritmo se cumple cuando no hay cambios en las etiquetas después de cierto número de iteraciones.

El algoritmo propuesto en [5], en el cual se basa esta investigación, utiliza *clustering* jerárquico para agrupar las imágenes. Previo al algoritmo de *clustering*, los autores aplican una función de mejora del ruido del sensor, que fortalece los componentes bajos y atenúa los componentes altos en el dominio wavelet, con la finalidad de eliminar los detalles de la escena en el mismo. Con una matriz de similitud que contiene todas las correlaciones entre los diferentes SPN y tomando como punto de partida a cada imagen como un *cluster* único, el algoritmo de *clustering* agrupa los dos *clusters* con un valor de correlación más alta formando un solo *cluster* y actualiza la matriz con una nueva fila y columna que vienen a sustituir las filas y columnas de los *clusters* agrupados. El criterio de enlace elegido para mezclar dos *clusters* fue el de enlace promedio. En cada iteración del algoritmo se almacena en una partición el estado de los *clusters* en ese momento y se calcula el coeficiente silueta global. Al final del algoritmo se elige la partición cuyo valor del coeficiente silueta sea el mínimo. En esa partición el número de *clusters* debería corresponderse con número de dispositivos que existen inicialmente, así como el contenido de cada *cluster* con los SPN de cada dispositivo. Los autores realizan una etapa de entrenamiento con el algoritmo descrito y una etapa de clasificación para las imágenes restantes. Para realizar esto basta obtener el promedio de los SPN por cada *cluster* y compararlos contra las imágenes restantes, la imagen se clasificará dentro del *cluster* cuya correlación sea más alta.

III. Descripción de la Técnica

El algoritmo de agrupación sin supervisión propuesto está basado en el presentado en [5]. Se trata de una combinación entre un *clustering* jerárquico y un *clustering* plano. Es decir, a pesar de formar una estructura de dendrograma con cada iteración del algoritmo, al final los *clusters* son tomados como entidades sin relación alguna ya que cada uno de ellos debe corresponder a un dispositivo específico.

Previo a realizar el *clustering*, es necesario obtener los patrones de ruido del sensor del conjunto de imágenes $I^{(i)},\ i = 1, ..., N$ utilizando el algoritmo de extracción y el parámetro de supresión de ruido $s_0 = 5$ propuestos en [14]:

$$n^{(i)} = I^{(i)} - F\left(I^{(i)}\right) \tag{2}$$

Donde n es el patrón de ruido de cada imagen i, I es el conjunto de imágenes con ruido del sensor y F es el filtro de extracción del ruido basado en la transformada wavelet. Para esto se utilizó el algoritmo desarrollado por Goljan et-al en [15]. En nuestra propuesta no se ha utilizado ningún algoritmo de mejoramiento de ruido, como los propuestos por [5] y [4]. El filtro de Wiener en el dominio de la frecuencia es suficiente para eliminar la mayoría de los detalles de la escena presentes al extraer el SPN.

Para cada uno de los N ruidos $(n_1, ..., n_N)$ se obtiene el valor de correlación usando la ecuación 1 y esto genera una matriz de similitud H de $N \times N$. Dicha matriz es simétrica y está compuesta de unos en su diagonal principal (ya que la correlación de un ruido consigo mismo es 1) . Una vez generada la matriz no será necesario volver a calcular las correlaciones entre ruidos a lo largo del algoritmo de *clustering* ahorrando tiempo y capacidad de procesamiento.

El algoritmo de *clustering* jerárquico seleccionado consiste en encontrar dentro de la matriz H el par de ruidos k y l con un valor de correlación más alto. Cabe mencionar que los valores de correlación en la diagonal principal no se toman en cuenta. A continuación las filas y columnas correspondiente a k y l son eliminadas y tanto una nueva fila como una nueva columna son agregadas a la matriz. Los valores de esta nueva fila y columna son el resultado de una función de criterio de enlace. La función elegida para este trabajo fue el criterio de enlace promedio puesto que sus resultados son más satisfactorios que con otros criterios de enlace como criterio simple o criterio completo, tal como se sugiere en [5]. La ecuación 3 muestra la función del criterio de enlace promedio entre dos *clusters* A y B.

$$H\left(A, B\right) = \frac{1}{\|A\| \, \|B\|} \sum_{ni \in A, nj \in B} corr\left(n_i, n_j\right) \tag{3}$$

Donde el valor $corr(n_i, n_j)$ se calcula con la ecuación 1 y puede ser tomado de la matriz H para simplificar el procesamiento computacional. $\|A\|$ y $\|B\|$ son la cardinalidad de los *clusters* A y B respectivamente.

Cada iteración del algoritmo toma los dos *clusters* con el valor de correlación más alto en la matriz y mezcla los objetos contenidos en éstos para crear un nuevo *cluster*, al mismo tiempo que almacena el estado de los distintos *clusters* en particiones $P_0, ..., P_{N-1}$ con el objetivo de conocer el contenido de los *clusters* en cada momento. En el *clustering* jerárquico, el resultado final del algoritmo es un *cluster* que contiene a todos los objetos. Sin embargo, en este trabajo para el agrupamiento de fotografías, cada *cluster* debería representar un dispositivo al final de la ejecución. Por este motivo se usó el coeficiente silueta como medida de validación de *clusters*. El coeficiente silueta mide el índice de similitud entre los elementos de un mismo *cluster* (cohesión) y la

similitud entre los elementos de un *cluster* con respecto a los demás (separación). A diferencia de Caldelli et al. [5] el cálculo del coeficiente silueta se realiza por cada *cluster* contenido en la partición P_i y no por cada patrón de ruido, como observamos en la ecuación 4.

$$s_j = \max(b_j) - a_j \tag{4}$$

donde, a_j (cohesión) es la correlación promedio entre todos los patrones de ruido dentro del *cluster* c_j. b_j (separación) es la correlación promedio de los patrones de ruido contenidos en el *cluster* c_j con respecto a los patrones de ruidos en los *clusters* restantes. Se toma el *cluster* vecino más cercano, es decir, aquel con la correlación más alta.

Para cada iteración q del algoritmo se obtiene una medida global de todos los coeficientes siluetas calculados a partir de los K *clusters*. Esto equivale a promediar los valores s_j en q. La ecuación 5 muestra dicho cálculo.

$$SC_q = \frac{1}{K}\sum_{j=1}^{K} s_j \tag{5}$$

Una vez concluido el *clustering* jerárquico se procede a buscar el SC_q con el valor mínimo, lo cual indica que los *clusters* de la partición P_q^* están en un nivel de correlación mayor. El número de *clusters* en ese instante debería corresponder al número real de dispositivos. El objetivo de almacenar la partición en cada momento del algoritmo es evitar volver a ejecutar el *clustering* ya que se tiene información de todos los *clusters* en cada iteración q.

En el Algoritmo 1 se muestra el pseudocódigo de la propuesta.

Algorithm 1: Algoritmo de clustering

1. Calcular el patrón de ruido $n^{(i)}$ de cada imagen donde $i \in 1, ..., N$;
2. Generar matriz de similitud $H \in R^{NxN}$;
3. **foreach** $q \in 1, ..., N-1$ **do**
4. Encontrar el par de *clusters* $H(k,l)$ con la mayor similitud;
5. Eliminar el par de filas y columnas correspondientes a los *clusters* k y l;
6. Calcular los valores del nuevo *cluster* usando el criterio de enlace promedio y agregar tanto la fila como columna correspondientes;
7. Determinar el coeficiente silueta global SC_q;
8. Almacenar la partición P_q;
9. Encontrar la partición donde el coeficiente silueta mínimo $min_q(SC_q)$;

IV. Experimentos y Resultados

Los experimentos fueron realizados con un conjunto total de 1050 fotografías de 7 modelos diferentes de cámaras de dispositivos móviles (Apple iPhone 5, Huawei U8815, Nokia 800 Lumia, Samsung GT-S5830M, LG E400, Sony ST25a y Zopo ZP980). Del conjunto total hay 150 fotografías de cada modelo.

Todas las imágenes fueron recortadas a 1024x1024 píxeles, poseen una orientación horizontal y son tanto de interiores como de exteriores con el objetivo de simular un escenario más realista. En la extracción del patrón de ruido de todas las imágenes se utilizó el promedio a cero (zero-mean) de filas y columnas, los 3 canales de color RGB fueron convertidos a una sola matriz de intensidades de grises, eliminando la información correspondiente al tono y la saturación, pero conservando la luminancia.

Para medir el grado de certeza en los resultados se utilizó la tasa de verdaderos positivos TPR (*True Positive Rate*). El TPR promedio para cada uno de los siguientes experimentos se calcula, computando para cada *cluster* el número de fotos que han sido bien clasificadas (TPR de cada *cluster*) y promediando los TPR de todos los *clusters* resultantes (si hay menos *clusters* que dispositivos se promedia teniendo en cuenta el número de dispositivos). Para calcular el TPR de cada *cluster*, hay que detectar en el *cluster* cual es el dispositivo que tiene el mayor número de imágenes con respecto al total de imágenes por dispositivo, siendo ese el *cluster* predominante del dispositivo, posteriormente hay que calcular el porcentaje de fotos que ha sido bien clasificadas para ese dispositivo en ese *cluster*. Realmente en la inmensa mayoría de los casos puede verse fácilmente que un *cluster* se asocia a uno o varios dispositivos como puede apreciarse en matrices de confusión de las Tablas I, II y III. Si hay varios *clusters* con el mismo número de fotos de un dispositivo o un *cluster* con igual número de fotos de varios dispositivos y a su vez éstos son los máximos, se toma como *cluster* predominante para el dispositivo el que se desee de entre las distintas opciones. Puede darse el caso que si hay un *cluster* de más, un *cluster* no sea predominante de ningún dispositivo (ver Tabla II) y su TPR para ese *cluster* sea 0. También puede que se forme un *cluster* menos (ver Tabla III), en este caso este se tendrá en cuenta la asociación del *cluster* al dispositivo y utilizar para el promedio el número de dispositivos como se indicó anteriormente.

Tabla I
TPR CON IGUAL NÚMERO DE DISPOSITIVOS QUE CLUSTERS

Marca - Modelo	Clusters					TPR
	1	2	3	4	5	promedio
Apple Iphone 5	49	0	0	1	0	
Huawei U8815	0	50	0	0	0	
LG E400	0	1	49	0	0	
Nokia 800 Lumia	0	0	0	50	0	
Samsung GT5830m	0	0	0	0	50	
TPR por *cluster*	98 %	100 %	98 %	100 %	100 %	99.2 %

En los resultados de los experimentos se consideran 3 posibles casos: a) Número de *clusters* identificados igual al número de dispositivos, b) número de *clusters* identificados mayor al número de dispositivos, y c) número de *clusters* identificados menor al número de dispositivos. Aunque el

Tabla II
TPR CON MENOR NÚMERO DE DISPOSITIVOS QUE CLUSTERS

Marca - Modelo	Clusters				TPR
	1	2	3	4	promedio
Apple I- phone 5	100	0	0	0	
Huawei - U8815	0	100	0	0	
LG - E400	0	0	97	3	
TPR por *cluster*	100 %	100 %	97 %	0 %	99 %

Tabla III
TPR CON MAYOR NÚMERO DE DISPOSITIVOS QUE CLUSTERS

Marca - Modelo	Clusters				TPR
	1	2	3	4	promedio
Apple Iphone 5	100	0	0	0	
Huawei U8815	0	100	0	0	
LG E400	0	0	100	0	
Nokia 800 Lumia	100	0	0	0	
Samsung GT 5830M	0	0	0	100	
TPR por *cluster*	100 %	100 %	100 %	100 %	80 %

primer caso es el ideal, el segundo caso también tiene un valor alto de TPR puesto el algoritmo deja ciertas imágenes desasociadas, es decir, fuera del *cluster* que les corresponde y estas no son consideradas en el TPR por no ser acierto. Por otro lado, en el último caso es donde se tienen porcentajes de acierto más bajo porque el algoritmo une dos o más dispositivos dentro de un mismo *cluster*.

Se realizaron varios experimentos para comparar los resultados entre recortar la imagen desde el centro o desde la esquina superior izquierda, teniendo este último criterio un TPR más alto. Una de las posibles razones por la que las dos zonas de recortes obtienen distintos resultados, es porque generalmente, las fotografías se toman enfocando en el centro el objeto de interés, el cual normalmente tiene un mayor grado de detalle. Este alto grado de detalle en el recorte de la imagen, en ciertos casos, puede dificultar la clasificación de la misma. La Tabla IV muestra el TPR en función del número distinto de dispositivos utilizados y el número de fotos utilizadas por dispositivo. Todos los dispositivos tienen el mismo número de fotos. En la Tabla IV se puede observar como el TPR aumenta en el caso del recorte en el centro a medida que se agrupan más dispositivos mientras que en el recorte por la esquina se mantienen buenos resultados.

Tabla IV
TPR EN FUNCIÓN DEL NÚMERO DISTINTO DE DISPOSITIVOS Y EL NÚMERO DE FOTOS POR DISPOSITIVO

Número de Fotos	Crop Corner			Crop Center		
	Número de Dispositivos			Número de Dispositivos		
	3	5	7	3	5	7
50	99.33 %	99.20 %	99.71 %	66.67 %	80 %	99.71 %
100	99 %	100 %	99.57 %	66.67 %	80 %	99.71 %

En un escenario cerrado no es muy probable contar con el mismo número de imágenes de cada dispositivo a identificar, por esa razón se realizaron experimentos en donde los conjuntos de imágenes por cada dispositivo no poseen una distribución simétrica para comprobar la adaptabilidad del algoritmo propuesto en un escenario real. En las Tablas V y VI se presentan los resultados obtenidos de agrupar las imágenes de 5 y 7 dispositivos respectivamente. El número de imágenes por dispositivo es variado y aún así podemos observar un muy alto grado de acierto (97.76 % TPR promedio de los experimentos de las Tablas V y VI).

Como se puede observar en los casos de número de imágenes asimétrico se ha experimentado con grupos de bastante disparidad numérica y en algunos casos con grupos pequeños (5 imágenes de un tipo de dispositivo), aun así se han logrado resultados de agrupación satisfactorios. Cabe destacar que existe una diferencia significativa en el resultado del experimento del grupo C de la Tabla VI, ya que se obtiene un TPR sensiblemente más bajo que el obtenido en el resto de experimentos. La causa a esta situación es que como se puede observar este experimento hay una cámara (Zopo Zp980) con una sola imagen. El hecho de que haya una sola imagen de un dispositivo hace que exista una alta probabilidad de que ese *cluster* no se genere correctamente, ya que sólo existen dos casos, la generación correcta al 100 % o la fusión de este *cluster* con otro. Concretamente en este experimento la imagen del Zopo Zp980 no se ha clasificado correctamente como *cluster* independiente y se ha fusionado con el *cluster* del Huawei U8815, bajando considerablemente el TPR. Simplemente para este experimento si se hubiera clasificado en un *cluster* independiente esa única imagen, el TPR habría sido del 99,71 %. Como puede observarse para *clusters* con una única imagen si se da una clasificación incorrecta el TPR baja sensiblemente, ya que esa única imagen hace que el TPR parcial del *cluster* sea del 0 %, aunque la práctica totalidad de las imágenes se hayan clasificado correctamente.

Tabla V
TPR PARA CLUSTERING ASIMÉTRICO DE 5 DISPOSITIVOS

Grupo	Apple Iphone 5	Huawei U8815	LG E400	Nokia 800 Lumia	Samsung GT 5830m	TPR
A	100	95	90	85	80	99.78 %
B	50	45	40	35	30	99.1 %
C	100	75	50	25	10	99.6 %
D	100	30	20	10	5	99 %

Tabla VI
TPR PARA CLUSTERING ASIMÉTRICO DE 7 DISPOSITIVOS

Grupo	Apple Iphone 5	Huawei U8815	LG E400	Nokia 800 Lumia	Samsung GT 5830m	Sony ST25a	Zopo Zp980	TPR
A	100	95	90	85	80	75	70	99.84 %
B	50	45	40	35	30	25	20	99.36 %
C	100	75	50	25	10	5	1	85.43 %
D	100	50	40	30	20	10	5	99.21 %

V. CONCLUSIONES

En este trabajo se ha realizado un análisis de las principales técnicas de agrupación de imágenes sin supervisión, siendo estas de suma importancia en el análisis forense de imágenes

digitales. A pesar del auge que han tenido las cámaras de dispositivos móviles en estos tiempos, aún no existen en el estado del arte muchas referencias para la agrupación no supervisada de imágenes de dispositivos móviles. La mayor parte de los trabajos se refieren a la clasificación supervisada y en muchos casos no se centran en imágenes de dispositivos móviles, las cuales tienen características peculiares. La comparación de los resultados de este trabajo con los de otros trabajos del estado del arte no puede realizarse de forma precisa, ya que en los mismos no se hace referencia al número final de *clusters* generados, lo cual es un tema fundamental. Además en estos trabajos no se detalla como se han calculado las tasas de acierto, ni se hace referencia a las mismas cuando los *clusters* generados por la clasificación son diferentes en número a la cantidad dispositivos utilizados, haciendo esto que la comparativa de sus tasas con respecto a nuestra interpretación del TPR carezca de sentido. El ruido agregado en cada fotografía por el sensor de la cámara, debido a las fallos en el proceso de fabricación de este o defectos por el uso diario, ha demostrado ser una fuente fiable de identificación de un dispositivo. Asimismo, el cálculo de correlación normalizada entre ruidos de sensor extraídos de dos o más fotografías es una medida de similitud bastante utilizada en las técnicas de aprendizaje sin supervisión de imágenes, siendo las técnicas de *clustering* aquellas que tienen mejores resultados.

El algoritmo de esta propuesta está basado en la combinación de un *clustering* jerárquico y un *clustering* plano para la separación entre *clusters*. El uso del coeficiente silueta para la validación de los *clusters* demostró dar buenos resultados al obtener elevados TPR, también el número de *clusters* correspondió al número de dispositivos reales en la mayoría de los casos.

El porcentaje de aciertos al utilizar el recorte de la imagen desde la esquina izquierda era más estable que aquellos recortados por el centro, pese a encontrar diferentes observaciones en la literatura argumentando la saturación y ausencia de iluminación encontrada en esas regiones.

Los experimentos realizados en este trabajo han permitido comprobar gran diversidad de situaciones con respecto a la simetría o no de los conjuntos del fotos, el tamaño de los mismos, el número de dispositivos utilizados y el uso de dispositivos de la misma marca. Tras todos los experimentos realizados se concluye que los resultados de la aplicación de la técnica son buenos (98.01 % TPR promedio de todos los experimentos realizados).

AGRADECIMIENTOS

El Grupo de Investigación GASS agradece la infraestructura proporcionada por el Campus de Excelencia Internacional (CEI) Campus Moncloa - Clúster de Cambio Global y Nuevas Energías (y, más concretamente, el sistema EOLO como recurso de computación de alto rendimiento HPC - High Performance Computing), infraestructura financiada por el Ministerio de Educación, Cultura y Deporte (MECD) y por el Ministerio de Economía y Competitividad (MINECO).

REFERENCIAS

[1] A. L. Sandoval Orozco, D. M. Arenas González, J. Rosales Corripio, L. J. García Villalba, and J. C. Hernandez-Castro, "Techniques for Source Camera Identification," in *Proceedings of the 6th International Conference on Information Technology*, May 2013, pp. 1–9.

[2] T. Van Lanh, K. S. Chong, S. Emmanuel, and M. S. Kankanhalli, "A Survey on Digital Camera Image Forensic Methods," in *Proceedings of the IEEE International Conference on Multimedia and Expo*. IEEE, July 2007, pp. 16–19.

[3] J. Fridrich, "Digital Image Forensics," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 26–37, March 2009.

[4] C.-T. Li, "Unsupervised Classification of Digital Images Using Enhanced Sensor Pattern Noise," in *Proceedings of the IEEE International Symposium on Circuits and Systems*. IEEE, May 2010, pp. 3429–3432.

[5] R. Caldelli, I. Amerini, F. Picchioni, and M. Innocenti, "Fast Image Clustering of Unknown Source Images," in *Proceedings of the IEEE International Workshop on Information Forensics and Security*. IEEE, December 2010, pp. 1–5.

[6] L. Rokach, "A Survey of Clustering Algorithms," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Springer US, 2010, pp. 269–298.

[7] C. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Transactions on Computers*, vol. C-20, no. 1, pp. 68–86, January 1971.

[8] J. D. Banfield and A. E. Raftery, "Model-Based Gaussian and Non-Gaussian Clustering," *Biometrics*, vol. 49, no. 3, pp. pp. 803–821, September 1993.

[9] D. Fisher, "Knowledge Acquisition Via Incremental Conceptual Clustering," *Machine Learning*, vol. 2, no. 2, pp. 139–172, 1987.

[10] J. Vesanto and E. Alhoniemi, "Clustering of the Self-Organizing Map," *IEEE Transactions on Neural Networks*, vol. 11, no. 3, pp. 586–600, May 2000.

[11] F. Hoppner, *Fuzzy Cluster Analysis: Methods for Classification, Data Analysis and Image Recognition*, ser. Jossey-Bass higher and adult education series. Wiley, 1999.

[12] S. Z. Selim and K. Alsultan, "A Simulated Annealing Algorithm for the Clustering Problem," *Pattern Recogn.*, vol. 24, no. 10, pp. 1003–1008, Oct. 1991.

[13] B.-b. Liu, H.-K. Lee, Y. Hu, and C.-H. Choi, "On Classification of Source Cameras: A Graph Based Approach," in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS),*. IEEE, December 2010, pp. 1–5.

[14] J. Lukas, J. Fridrich, and M. Goljan, "Digital Camera Identification from Sensor Pattern Noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, June 2006.

[15] M. Goljan, J. Fridrich, and T. Filler, "Large Scale Test of Sensor Fingerprint Camera Identification," in *Proceedings of the SPIE on Media Forensics and Security*, vol. 7254. International Society for Optics and Photonics, February 2009, pp. 72 540I–72 540I.

Identificación de la Fuente de Imágenes de Dispositivos Móviles Basada en el Ruido del Sensor

Jocelin Rosales Corripio, David Manuel Arenas González, Ana Lucila Sandoval Orozco,
Luis Javier García Villalba

Grupo de Análisis, Seguridad y Sistemas (GASS), Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid
Email: jocerosa@ucm.es, {darenas, asandoval, javiergv}@fdi.ucm.es

***Resumen*—La fuente de una imagen digital se puede identificar a través de los rasgos que el dispositivo que la genera impregna en ella durante el proceso de su generación. La mayoría de las investigaciones realizadas en los últimos años sobre técnicas de identificación de fuente se han enfocado únicamente en la identificación de cámaras tradicionales DSC (*Digital Still Camera*). Considerando que hoy en día las cámaras de los dispositivos móviles prácticamente han sustituido a las DSCs se detectó la necesidad de realizar investigación sobre las técnicas para identificar la fuente de imágenes generadas por dispositivos móviles. Las imágenes digitales generadas por una cámara digital contienen intrínsecamente un patrón del ruido del sensor que se puede usar como medio de identificación de la fuente. Específicamente, las cámaras digitales de dispositivos móviles cuentan en su mayoría con un tipo de sensor que deja rasgos característicos en la imagen. En este trabajo se propone un algoritmo basado en el ruido del sensor y en la transformada wavelet para identificar el dispositivo móvil (marca y modelo) que ha generado determinadas imágenes bajo investigación.**

***Palabras clave*—Análisis forense, imagen digital, patrón de ruido del sensor, PRNU. *(Forensics analysis, digital image, sensor pattern noise, PRNU)*.**

I. Introduction

Con frecuencia las fotografías son consideradas como una parte de la verdad al ser hechos reales capturados por dispositivos electrónicos (cámaras). Sin embargo, con el desarrollo de la tecnología han surgido herramientas potentes y sofisticadas que facilitan de una manera impresionante la alteración de las imágenes digitales, incluso para quienes no tienen conocimientos técnicos o especializados en el área [1].

El desarrollo de las tecnologías digitales ha estado y continúa avanzando a un ritmo imparable. Cada día el número de cámaras digitales va creciendo, así como la facilidad de acceso a ellas. Las cámaras digitales de móviles merecen especial atención, ya que estudios realizados indican que al final del año 2012 el número total de dispositivos móviles activos alcanzó los 6,7 billones y se estima que para el verano del 2013 este número igualará al total de la población del planeta 7,1 billones. El 83 % de estos dispositivos móviles cuentan con cámara digital integrada, las cuales a diferencia de las cámaras digitales convencionales son llevadas por sus dueños todo el tiempo a la mayoría de lugares que asiste y en muchos casos tienen conexión a internet [2].

Debido al incremento en sus capacidades de almacenamiento, procesamiento, usabilidad y portabilidad así como a su bajo coste, los dispositivos móviles están presentes en diversidad de actividades, lugares y eventos de la vida diaria. A causa del extenso uso de las cámaras digitales de dispositivos móviles se han generado polémicas, discusiones y normas sobre la prohibición de su uso en lugares como escuelas, oficinas de gobierno, eventos empresariales, conciertos, empresas, etc. Una consecuencia más de su extenso uso es que las imágenes digitales en la actualidad son utilizadas como testigos silenciosos en procesos judiciales, siendo una pieza crucial de la evidencia del crimen [3]. Es por ello que contar con herramientas que permitan identificar a los dispositivos que han generado una cierta imagen digital cobra importancia ya que podría servir en diversas áreas como la lucha contra la pornografía infantil, la prevención de robo de tarjetas de crédito, el combate a la piratería, la prevención de secuestros, etc.

II. Técnicas de Análisis Forense en Imagenes

La investigación en este campo estudia el diseño de técnicas para identificar las características, especialmente marca y modelo, de los dispositivos utilizados para la generación de imágenes digitales. El éxito de estas técnicas depende del supuesto de que todas las imágenes adquiridas por un mismo dispositivo presentan características intrínsecas del dispositivo. Las características que se usan para identificar marca y modelo de las cámaras digitales se derivan de las diferencias que existen entre las técnicas de procesamiento de las imágenes y las tecnologías de los componentes que se utilizan [4]. El mayor problema con este enfoque es que los diferentes modelos de las cámaras digitales usan componentes de un número reducido de fabricantes, y que los algoritmos que usan también son muy similares entre modelos de la misma marca. Es por ello que la fiabilidad de la identificación de la cámara fuente depende en gran parte de la identificación de varias características independientes del modelo. Según [4] se pueden establecer cuatro grupos de técnicas para este fin: utilización de la aberración de las lentes, interpolación de la matriz CFA, uso de las características de la imagen e imperfecciones del sensor. Esta última constituye el objeto de

este trabajo. Además de las anteriores existe otro grupo de técnicas basadas en los metadatos.

Las técnicas basadas en el estudio de las huellas que los defectos del sensor dejan sobre las imágenes se dividen en dos ramas: defectos de píxel y patrón de ruido del sensor SPN (Sensor Pattern Noise). En la primera se estudian los defectos de píxel, los píxeles calientes, los píxeles muertos, los defectos de fila o columna, y los defectos de grupo. En la segunda se construye un patrón del ruido promediando los múltiples residuos de ruido obtenidos mediante algún filtro de eliminación de ruido. La presencia del patrón se determina utilizando algún método de clasificación como correlación o máquinas SVM.

En [5] se estudian los defectos de los píxeles en los sensores de tipo CCD, centrándose en la evaluación de diferentes características para examinar las imágenes e identificar la fuente: defectos del sensor CCD, formato de los archivos usados, ruido introducido en la imagen y marcas de agua introducidas por el fabricante de la cámara. Entre los defectos del sensor CCD considerados se encuentran los puntos calientes, los píxeles muertos, los defectos en grupo y los defectos de fila o columna. En sus resultados se observa que cada una de las cámaras tiene un patrón de defecto diferente. Sin embargo, también se señala que el número de defectos en los píxeles para una cámara es diferente entre fotos y varía demasiado en función del contenido de la imagen. Asimismo, se revela que el número de defectos cambia con la temperatura. Al considerar únicamente los defectos de los sensores de tipo CCD este estudio no es aplicable al análisis de imágenes generadas por dispositivos móviles.

En [6] se analiza el patrón de ruido del sensor de un conjunto de cámaras, el cual funciona como una huella dactilar, permitiendo la identificación única de cada cámara. Para obtener este patrón se realiza un promedio del ruido obtenido a partir de diferentes imágenes utilizando un filtro de eliminación de ruido. Para identificar la cámara a partir de una imagen dada, se considera el patrón de referencia como una marca de agua cuya presencia en la imagen es establecida mediante un detector de correlación. El estudio se realizó con 320 imágenes procedentes de 9 modelos distintos de cámaras. También se demuestra que este método está afectado por algoritmos de procesamiento de la imagen como la compresión JPEG y la corrección gamma. Los resultados para fotografías con diferentes tamaños y recortadas no son satisfactorios [4].

En [7] se propone un enfoque para la identificación de la cámara fuente considerando escenarios abiertos, donde a diferencia de los escenarios cerrados no se da por sentado contar con acceso a todas las posibles cámaras de origen de la imagen. Este enfoque, considera 9 diferentes áreas de interés ROI (*Region Of Interest*) que se encuentran en las esquinas y el centro de las imágenes. El uso de las regiones de interés permite trabajar con imágenes de diferentes resoluciones sin la necesidad de rellenar con ceros las imágenes y sin el uso de artefactos de interpolación de color. Para determinar las características se calcula el SPN para cada uno de los canales R, G y B. Asimismo, se calcula el SPN para el canal Y (luminancia), generándose un total de 36 características para representar cada imagen. Después, las imágenes tomadas por la cámara bajo investigación son etiquetadas como la clase positiva y las tomadas por las cámaras disponibles restantes como las clases negativas. Después de la fase de entrenamiento de la SVM en la que se calcula el hiper-plano que separa los casos positivos y negativos toman en cuenta las clases desconocidas del escenario abierto moviendo el hiper-plano generado por un valor dado ya sea hacia adentro (hacia las clases positivas) o hacia afuera (las clases negativas). En los experimentos utilizan un conjunto de 25 cámaras digitales de 9 fabricantes, 150 imágenes en formato JPEG de cada cámara con diferentes configuraciones de luz, zoom y flash. Los resultados de los experimentos mostraron una precisión del 94,49 %, del 96,77 % y del 98,10 %, utilizando conjuntos abiertos con 2/25, 5/25, y 15/25 cámaras, respectivamente, definiendo un conjunto abierto x/y como el conjunto de y cámaras donde x cámaras son usadas para entrenar y probar las imágenes que pueden pertenecer a cualquiera de las cámaras x conocidas, así como a las otras y-x cámaras desconocidas.

En [8] se basan en el trabajo de [6] para extraer el ruido del sensor usando el cálculo de similitudes como método de la clasificación. Exponen que el ruido del sensor puede estar muy contaminado por los detalles de los escenarios y proponen que entre más fuerte es un componente del ruido del sensor es menos fiable y por lo tanto debe ser atenuado. Proponen una forma de atenuar los valores altos del ruido del sensor y realizan experimentos de identificación con 6 cámaras tradicionales diferentes (100 imágenes de cada cámara). Para las imágenes de 1536x2048 pixeles obtuvieron una tasa de acierto del 38.5 % con la implementación sin la mejora y del 80.8 % con la mejora propuesta; para las imágenes de 512x512 píxeles obtuvieron una tasa de acierto del 21.8 % sin la mejora y del 78.7 % con la mejora propuesta.

III. Algoritmo de Identificación de la Fuente

Debido a la propiedad determinista del patrón de ruido del sensor que está presente en cada imagen capturada, se puede usar este patrón como huella para identificar el dispositivo que generó la imagen objeto en investigación. Haciendo una analogía, se puede decir que el patrón del ruido del sensor es para una cámara digital lo que la huella para un ser humano.

Para poder identificar la marca y el modelo de la cámara digital de un dispositivo móvil se requiere de un algoritmo que nos permita extraer el ruido del sensor y otro que nos permita obtener las características de las huellas obtenidas para así poder clasificarlas e identificarlas.

Tomando como referencia las ideas principales de [6] se propone un algoritmo para extraer el ruido del sensor (también conocido como ruido residual) que se describe en el algoritmo 1.

Con el promediado a cero se limpia la huella de las características que no son intrínsecas al sensor aplicando como se sugiere en [9], de tal manera que los promedios de las filas y de las columnas sean iguales a cero. Esto se logra restando el promedio de la columna a cada píxel de la columna y

Algoritmo 1: Extraer ruido del sensor

Input: Imagen
varianza: (adaptativa o no adaptativa)
Result: Huella del sensor de la imagen

1 **procedure** EXTRAERHUELLA(I)
2 Realizar descomposición wavelet de 4 niveles de I_n;
3 **foreach** *nivel de la descomposición wavelet* **do**
4 **foreach** $c \in \{H,V,D\}$ **do**
5 Calcular la varianza local;
6 **if** *varianza adaptativa* **then**
7 Calcular 4 varianzas con ventanas de tamaños 3, 5 ,7 y 9 respectivamente;
8 Seleccionar la varianzas mínima;
else
9 Calcular la varianza con una ventana de tamaño 3;
10 Calcular los componentes wavelet sin ruido aplicando el filtro de Wiener a la varianza;
11 Obtener la imagen limpia del ruido del sensor aplicando la Transformada Inversa Wavelet;
12 Calcular el ruido del sensor con $I_{ruido}=I_{entrada}-I_{limpia}$;
13 Aplicar a I_{ruido} un promediado a cero;
14 Aumentar en I_{ruido} el peso del canal verde con $I_{ruido}=0{,}3\cdot I_{ruido_R}+0{,}6\cdot I_{ruido_G}+0{,}1\cdot I_{ruido_B}$;
15 **end procedure**

Algoritmo 2: Extracción de Características

Input: Imagen
Huella del sensor de la imagen
Result: 81 características

1 **procedure** EXTRAERCARACTERISTICAS(I)
2 Separar los canales R, G y B de la huella del sensor;
3 **foreach** *canal de color* **do**
4 Hacer una descomposición wavelet de un nivel;
5 **foreach** $c \in \{H,V,D\}$ **do**
6 Calcular k momentos centrales con $m_k = \frac{1}{n}\sum_{i=1}^{n} |c_i - \overline{c}|^k$;
7 **end procedure**

posteriormente restando el promedio de la fila a cada píxel de la fila. Esta operación se aplica a todas las filas y columnas de la imagen. Después de limpiar la imagen se le da un mayor peso al canal verde ya que debido a la configuración de la matriz de color éste contiene más información sobre la imagen que el resto de los canales de color [10][11]. La identificación de las cámaras se realiza utilizando una máquina de soporte vectorial SVM para lo que es necesario extraer una serie de características que representen a las huellas de los sensores. Se calculan un total de 81 características (3 canales x 3 componentes wavelet x 9 momentos centrales) mediante el algoritmo 2.

Con las características que se extraen tanto de las imágenes para entrenamiento como para probar se alimenta la máquina SVM y se obtienen las clasificaciones.

IV. EXPERIMENTOS Y RESULTADOS

Para evaluar la efectividad del algoritmo de identificación de la fuente de dispositivos móviles se realizaron dos experimentos, en los que se consideraron los 1024x1024 píxeles centrales de las fotografías como se recomienda ampliamente en [12]. La Tabla I resume los principales parámetros utilizados.

En el primer experimento se probó con un grupo de 8 cámaras digitales de dispositivos móviles de 4 fabricantes. De Apple se consideraron los modelos iPhone3G (A1), iPhone4S (A2) y iPhone3 (A3); de BlackBerry el 8520 (B1); de Sony Ericsson el UST25a (SE1) y el U5I (SE2); y de Samsung el GTI9100 (S1) y el GTS5830 (S2). El algoritmo propuesto obtuvo un porcentaje de acierto promedio de 93.625 % al identificar entre marca y modelo como se observa en la matriz de confusión de la Tabla II.

Tabla I
PARÁMETROS UTILIZADOS EN LOS EXPERIMENTOS

Parámetro	Valor
Tipo de Fotos	Sin ninguna restricción
Dimensiones	1024 x1024
Fotos Entrenadas x Cámara	100
Fotos Probadas x Cámara	100
Cálculo de la Varianza	Enfoque no adaptativo

Tabla II
MATRIZ DE CONFUSIÓN DEL EXPERIMENTO 1

Cámara	A1	A2	A3	B1	SE1	SE2	S1	S2
A1	**92**	1	0	0	0	1	0	6
A2	0	**96**	0	0	1	0	3	0
A3	0	0	**99**	0	0	0	1	0
B1	0	0	0	**94**	0	2	0	4
SE1	7	2	0	0	**91**	0	0	0
SE2	2	0	0	1	0	**94**	1	2
S1	4	8	0	0	0	5	**83**	0
S2	0	0	0	0	0	0	0	**100**

Con la finalidad de acercarse a escenarios más reales el segundo experimento se realizó con 14 cámaras digitales de dispositivos móviles de 7 fabricantes. De Apple se consideraron los modelos iPhone3G (A1), iPhone4S (A2), iPhone3 (A3) y iPhone5 (A4); de BlackBerry el 8520 (B1); de Sony Ericsson el UST25a (SE1) y el U5I (SE2); de Samsung el GTI9100 (S1), el GTS5830 (S2) y el GT-S5830M (S3); de Lg el E400 (L1); de HTC el DesireHD (H1) y el Desire (H2); y de Nokia el E61I (N1). El algoritmo propuesto obtuvo un porcentaje de acierto promedio de 87,214 % como se puede observar en la matriz de confusión de la Tabla III.

Tabla III
MATRIZ DE CONFUSIÓN DEL EXPERIMENTO 2

Cámara	A1	A2	A3	A4	B1	SE1	SE1	S1	S1	S3	L1	H1	H2	N1
A1	**90**	0	0	2	0	0	0	0	7	0	1	0	0	0
A2	0	**91**	0	3	0	0	0	3	0	0	0	1	2	0
A3	0	0	**98**	0	0	0	0	2	0	0	0	0	0	0
A4	0	0	1	**88**	0	0	0	0	0	0	3	6	0	2
B1	0	0	0	2	**73**	0	0	0	4	0	0	1	0	20
SE1	7	0	0	0	0	**80**	0	0	0	0	1	12	0	0
SE2	1	0	0	2	2	0	**86**	1	2	5	1	0	0	0
S1	4	5	0	4	0	0	1	**83**	0	0	1	0	2	0
S2	0	0	0	0	0	0	0	0	**100**	0	0	0	0	0
S3	0	0	1	0	0	0	8	0	0	**85**	0	1	0	5
L1	0	0	0	9	0	6	0	0	2	0	**70**	13	0	0
H1	2	0	0	0	0	11	0	0	1	0	1	**85**	0	0
H2	0	6	0	0	0	0	0	0	0	0	0	0	**94**	0
N1	0	0	0	0	2	0	0	0	0	0	0	0	0	**98**

V. CONCLUSIONES

En este trabajo se estudian las diferentes técnicas de análisis forense de imágenes para solucionar el problema de la identificación de la fuente de una imagen. Se describe la idea principal de cada una de las técnicas así como algunos de los trabajos más representativos que se han realizado aplicándolas. De acuerdo a la estructura y funcionamiento de las cámaras digitales de dispositivos móviles las técnicas más adecuadas para realizar análisis forense en ellas son las que se basan en el ruido del sensor y las que utilizan las transformadas wavelet. En virtud de lo anterior se propuso un algoritmo para la identificación de los dispositivos móviles fuente combinando las técnicas basadas en la huella del sensor y en la transformación wavelet. Por último con los experimentos realizados y sus resultados se demuestra que la combinación de estas técnicas es efectiva para la identificación del modelo y fabricante con un alto porcentaje de acierto.

Aún estimando que son buenos los resultados obtenidos por la técnica, obviamente existe margen de mejora de las tasas de acierto, sobre todo teniendo en cuenta el caso en el que el número de cámaras aumenta considerablemente. Cuanto mayor sea la mejora en la tasa de acierto mayor será la posibilidad de aplicación de la técnica a situaciones reales. A grandes rasgos las principales líneas de investigación a tener en cuenta en los trabajos futuros son: mejora en la selección del recorte de la fotografía (distintas dimensiones y zonas), optimización de los parámetros de configuración de la máquina SVM, optimización en la selección de la función wavelet y la combinación de esta técnica con otras como las basadas en las caracterísitcas del color, las basadas en las métricas de calidad de la imagen o las que utilizan otros tipos de características extraídas del ruido del sensor.

REFERENCIAS

[1] T. Gloe, M. Kirchner, A. Winkler, and R. Bohme, "Can We Trust Digital Image Forensics?" in *Proceedings of the 15th International Conference on Multimedia*, September 2007, pp. 78–86.

[2] T. Ahonen and A. Moore, "Tomi Ahonen Almanac 2012: Mobile Telecoms Industry Annual Review," 2012. [Online]. Available: http://www.tomiahonen.com/ebook/almanac.html

[3] M. Al-Zarouni, "Mobile Handset Forensic Evidence: a Challenge for Law Enforcement," in *Proceedings of the 4th Australian Digital Forensics Conference*, December 2006.

[4] T. Van Lanh, K. S. Chong, S. Emmanuel, and M. S. Kankanhalli, "A Survey on Digital Camera Image Forensic Methods," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, July 2007, pp. 16–19.

[5] Z. J. Geradts, J. Bijhold, M. Kieft, K. Kurosawa, K. Kuroki, and N. Saitoh, "Methods for Identification of Images Acquired with Digital Cameras," in *Proceedings of the Enabling Technologies for Law Enforcement and Security Conference*, vol. 4232, February 2001, pp. 505–512.

[6] J. Lukas, J. Fridrich, and M. Goljan, "Digital Camera Identification from Sensor Pattern Noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, June 2006.

[7] F. D. O. Costa, M. Eckmann, W. J. Scheirer, and A. Rocha, "Open Set Source Camera Attribution," in *Proceedings of the 25th Conference on Graphics, Patterns and Images*, August 2012, pp. 71–78.

[8] C. T. Li, "Source Camera Linking Using eEnhanced Sensor Pattern Noise Extracted from Images," in *Proceedings of the 3rd International Conference on Crime Detection and Prevention (ICDP 2009)*. Curran Associates, Inc., December 2009, pp. 1–6.

[9] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining Image Origin and Integrity Using Sensor Noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, March 2008.

[10] O. Celiktutan, B. Sankur, and I. Avcibas, "Blind Identification of Source Cell-Phone Model," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 553–566, September 2008.

[11] C. McKay, "Forensic Analysis of Digital Imaging Devices," University of Maryland," Technical Report, 2007.

[12] C. T. Li and R. Satta, "On the Location-Dependent Quality of the Sensor Pattern Noise and its Implication in Multimedia Forensics," in *Proceedings of the 4th International Conference on Imaging for Crime Detection and Prevention 2011 (ICDP 2011)*. Curran Associates, Inc., November 2011, pp. 1–6.

Aprendizaje supervisado para el enlace de registros a través de la media ponderada

Daniel Abril
Instituto de Investigación
en Inteligencia Artificial (IIIA),
Consejo Superior de
Investigaciones Científicas (CSIC)
Email: dabril@iiia.csic.es

Guillermo Navarro-Arribas
Dep. Enginyeria de la Informació
i de les Comunicacions (DEIC),
Universitat Autònoma
de Barcelona (UAB)
Email: guillermo.navarro@uab.cat

Vicenç Torra
Instituto de Investigación
en Inteligencia Artificial (IIIA),
Consejo Superior de
Investigaciones Científicas (CSIC)
Email: vtorra@iiia.csic.es

***Resumen*—En el área de la privacidad de datos, las técnicas para el enlace de registros son utilizadas para evaluar el riesgo de revelación de un conjunto de datos protegido. La idea principal detrás de estas técnicas es enlazar registros que hacen referencia a un mismo individuo, entre diferentes bases de datos. En este trabajo se presenta una variación del enlace de registros basada en una media ponderada para calcular distancias entre registros. Mediante el uso de un método supervisado de aprendizaje nuestra propuesta permite determinar cuales son los pesos que maximizan el número de enlaces entre los registros de la base de datos original y su versión protegida. El resultado de este trabajo se aplica en la estimación del riesgo de revelación de datos protegidos.**

***Palabras clave*—enlace de registros (record linkage), privacidad de datos (data privacy), riesgo de divulgación (disclosure risk).**

I. Introducción

El enlace de registros consiste en el proceso de identificación de forma rápida y precisa de dos o más registros distribuidos en varias bases de datos (o fuentes de información en general) los cuales hacen referencia a la misma entidad o individuo. Esta técnica fue inicialmente introducida por Dunn [1] en el campo de la sanidad pública, con el fin de crear historiales médicos completos mediante el enlace de toda la información recopilada sobre un paciente, la cual estaba repartida por varias bases de datos. Estos enlaces fueron posibles gracias a la utilización de campos clave como el nombre o la fecha de nacimiento, entre otros. En los siguientes años, esta idea fue mejorada y matemáticamente formalizada [2], [3], [4]. Hoy en día se ha convertido en una popular técnica utilizada por agencias estadísticas, comunidades de investigación y otras instituciones, no solo para la integración de bases de datos [5], [6], sino también para la limpieza de datos [7] o el control de la calidad de los datos [8]. Un claro ejemplo de la utilización de estos métodos es para la detección de registros duplicados entre diferentes conjuntos de datos [9].

Debido a la necesidad de distintas agencias gubernamentales u otras instituciones de coleccionar y analizar grandes cantidades de datos confidenciales, las técnicas de enlace de registros fueron recientemente introducidas en el área de la privacidad de datos. Esta área de investigación proporciona métodos de seguridad para las bases de datos estadísticos con el fin de combatir la revelación de información confidencial contenida en dichas bases de datos. *Privacy Preserving Data Mining (PPDM)* [10] y *Statistical Disclosure Control (SDC)* [11] son dos disciplinas cuya función es la investigación de métodos y herramientas para asegurar la privacidad de estos datos. Dentro de estos campos el enlace de registros se utiliza para obtener una evaluación del riesgo de revelación de información confidencial sobre un conjunto de datos previamente protegido [12], [14]. Por lo tanto el riesgo de re-identificación de un individuo se evalúa mediante la identificación de enlaces de registros pertenecientes al mismo individuo entre los datos protegidos y originales. En [13] los autores definen un método general de evaluación de un conjunto de datos protegido basado en la combinación de diferentes medidas analíticas, cuyo objetivo es evaluar el riesgo de revelación de información confidencial y la evaluación de la cantidad de información perdida en el proceso de protección.

En este artículo se introduce un nuevo método de evaluación del riesgo de revelación con el fin de mejorar la precisión de las técnicas actualmente utilizadas. Este consiste en la utilización de una media ponderada como distancia en el proceso de enlace de registros y un algoritmo de aprendizaje supervisado, de manera que el algoritmo de aprendizaje aprenda cuales son los pesos que maximizan el número de enlaces entre el conjunto de datos original y el protegido.

La organización de este artículo es la siguiente. En la Sección II, se presentan algunos conceptos básicos necesarios para el resto de las secciones. En la Sección III se describe el método de aprendizaje supervisado para el enlace de registros cuando se utiliza la media ponderada. La evaluación del método presentado se introduce en la Sección IV. Finalmente, la Sección V presenta las conclusiones del trabajo realizado.

II. Enlace de registros en la privacidad de datos

En esta sección se presentan algunas ideas y definiciones básicas para comprender el uso de las técnicas de enlace de registros en el campo de la privacidad de datos.

PPDM y *SDC* están orientadas a trabajar sobre bases de datos, principalmente tablas o ficheros (*microdata*). Estas bases de datos pueden verse como una matriz, X, de N filas (registros) y n columnas (atributos), donde cada fila

corresponde a un único individuo (o entidad). En este contexto se pueden diferenciar dos tipos diferentes de atributos:

- *Identificadores*: son aquellos atributos que pueden identificar directamente un individuo, como por ejemplo el número de identificación nacional (DNI) o el número de cuenta bancaria.
- *Casi-identificadores*: son aquellos atributos que por ellos mismos no son capaces de identificar un único individuo, sin embargo, cuando dos o más de ellos son combinados, pueden identificar inequívocamente un individuo. Estos atributos se pueden dividir a su vez en dos tipos, los *confidenciales* (X_c) y los *no confidenciales* (X_{nc}), dependiendo del tipo de información que representen. Un ejemplo de atributo no confidencial sería el código postal, mientras que un ejemplo de atributo confidencial podría ser el salario.

Previa publicación de un conjunto de datos X, es necesaria su protección mediante la aplicación de un método ρ, el cual dará lugar a un conjunto de datos protegidos Y. Estos métodos de protección solo protegerán los atributos considerados como casi-identificadores no confidenciales, $Y_{nc} = \rho(X_{nc})$, ya que para asegurar la privacidad de los individuos los identificadores son eliminados y/o cifrados. Los casi-identificadores confidenciales no son modificados ni eliminados debido a su interés de análisis, por lo tanto quedarán intactos. De este modo, podemos ver el conjunto de datos protegido como $Y = \rho(X_{nc})||X_c$. Este escenario fue presentado en [13] y posteriormente utilizado en otros trabajos como [14].

En el campo de la privacidad de datos, el enlace de registros es utilizado para re-identificar individuos entre la base de datos protegida y la base de datos original, es decir, se usa como una medida de evaluación del riesgo de revelación. Actualmente existen dos enfoques diferentes del enlace de registros para la evaluación del riesgo. El *enlace de registros probabilístico (PRL)* [16] y el *enlace de registros basado en distancias (DBRL)* [17].

En el trabajo realizado en este artículo se utiliza el segundo tipo, el enlace de registros basado en distancias, el cual es explicado con detalle a continuación.

II-A. Enlace de registros basado en distancias

La idea principal del enlace de registros basado en distancias es la definición de una función de distancia. Como es sabido, dependiendo de la distancia utilizada se pueden obtener resultados completamente diferentes. A continuación se revisan dos de las distancias más utilizadas y testeadas en la literatura del DBRL, la distancia Euclídea y la distancia de Mahalanobis.

En este artículo usaremos $V_1^X, \ldots, V_n^X$ y $V_1^Y, \ldots, V_n^Y$ para indicar el conjunto de atributos de los ficheros X e Y, respectivamente. Usando esta notación, podemos expresar los valores de cada atributo de un registro $a \in X$ como $a = (V_1^X(a), \ldots, V_n^X(a))$ y un registro $b \in Y$ como $b = (V_1^Y(b), \ldots, V_n^Y(b))$. Además, indicaremos la media de los valores de un atributo V_i^X como $\overline{V_i^X}$.

- La *distancia Euclídea* es usada para bases de datos con atributos estandarizados. La distancia entre dos registros a y b se define como:

$$d(a,b)^2 = \sum_{i=1}^{n} \left(\frac{V_i^X(a) - \overline{V_i^X}}{\sigma(V_i^X)} - \frac{V_i^Y(b) - \overline{V_i^Y}}{\sigma(V_i^Y)} \right)^2 \quad (1)$$

- La *distancia de Mahalanobis* se define como:

$$dMD(a,b)^2 = (a-b)'\Sigma^{-1}(a-b)$$

donde, $\Sigma = [Var(V^X) + Var(V^Y) - 2Cov(V^X, V^Y]$ y $Var(V^X)$ es la varianza de los atributos V^X, $Var(V^Y)$ es la varianza de los atributos V^Y y $Cov(V^X, V^Y)$ es la covarianza entre los atributos V^X y V^Y. Si la matriz de covarianza es una matriz identidad, entonces la distancia de Mahalanobis se reduce a la distancia Euclídea.

III. Aprendizaje supervisado para el enlace de registros

En esta sección se presenta el método de aprendizaje supervisado, el cual se usa junto a una distancia ponderada, para determinar cuales son los pesos de esta distancia que maximizan el número de re-identificaciones entre los registros del fichero original y protegido. En la Sección III-A se introduce la distancia usada, la media ponderada, mientras que en la Sección III-B se introduce el problema de aprendizaje como un problema de optimización en base a la media ponderada.

III-A. Distancia ponderadas

Es bien conocido que la multiplicación de la distancia Euclídea por una constante no altera los resultados de ningún algoritmo de enlace de registros. De modo que se puede expresar la Ecuación 1 como una media ponderada. Esta se pude formalizar como:

$$d(a,b)^2 = \sum_{i=1}^{n} \frac{1}{n} \left(\frac{V_i^X(a) - \overline{V^X}_i(a)}{\sigma(V_i^X)} - \frac{V_i^Y(b) - \overline{V^Y}_i(b)}{\sigma(V_i^Y)} \right)^2$$

La cual, si definimos para cada atributo:

$$d_i(a,b)^2 = \left(\frac{V_i^X(a) - \overline{V^X}_i(a)}{\sigma(V_i^X)} - \frac{V_i^Y(b) - \overline{V^Y}_i(b)}{\sigma(V_i^Y)} \right)^2$$

Esta expresión se pude redefinir como,

$$d(a,b)^2 = AM(d_1(a,b)^2, \ldots, d_n(a,b)^2),$$

donde AM es la media aritmética

$$AM(c_1, \ldots, c_n) = \sum_i c_i/n$$

.

En general, cualquier operación de agregación $\mathbb{C}$ [18] puede utilizarse: $d(a,b)^2 = \mathbb{C}(d_1(a,b)^2, \ldots, d_n(a,b)^2)$.

De esta definición, es trivial la consideración de diferentes operadores de agregación, como por ejemplo la media ponderada.

Definición 1 *Considerando $p = (p_1, \ldots, p_n)$ como un vector de pesos, es decir, $p_i \geq 0$ y $\sum_i p_i = 1$. Entonces, la distancia ponderada se define como:*

$$d^2WM_p(a,b) = WM_p(d_1(a,b)^2, \ldots, d_n(a,b)^2),$$

donde $WM_p(c_1, \ldots, c_n) = \sum_i p_i \cdot c_i$.

III-B. Aprendizaje supervisado

Para simplificar la formalización del proceso, se asume que cada registro (fila) b_i de Y es la versión protegida de a_i de X. Es decir, los dos conjuntos de datos están alineados. Entonces, dos registros están correctamente enlazados usando un operador de agregación $\mathbb{C}$ cuando la agregación de los valores a_i y b_i es más pequeña que para a_i y b_j para todo $i \neq j$. Formalmente, a_i se considera correctamente enlazado con b_i cuando se satisface la siguiente ecuación para todo $i \neq j$.

$$\mathbb{C}(a_i, b_i) < \mathbb{C}(a_i, b_j) \tag{2}$$

En condiciones óptimas estas desiguales las satisfacen todos los registros a_i. No obstante, en general no se pude esperar que se cumpla debido a los errores introducidos en el conjunto de datos por el algoritmo de protección. Por lo tanto, el proceso de aprendizaje es formalizado como un problema de optimización.

La Ecuación (2) debe ser relajada de manera que la solución pueda violar alguna de las restricciones. Esta relajación es formalizada creando el concepto de bloque. Un bloque es el conjunto de ecuaciones referentes a un registro a_i, es decir, el conjunto de todas las distancias entre un registro original y todos los registros protegidos. Así, podemos asignar a cada bloque una variable K_i, teniendo tantas variables como número de registros. Además, hemos considerado para la formalización una constante C que multiplica K_i para superar las inconsistencias y satisfacer las restricciones. La idea de este enfoque es que cada variable K_i indique, por cada bloque, si todas las restricciones correspondientes se satisfacen ($K_i = 0$) o si por el contrario, no se satisfacen ($K_i = 1$). Así, si un registro a_i no cumple la Ecuación (2) por algún registro b_j, no importa que otro registro b_k ($k \neq j$) también viole la ecuación para el mismo a_i. Teniendo en cuenta esta asunción, el objetivo del problema será minimizar el número de bloques que no cumplen sus restricciones. De este modo, podremos encontrar los pesos que minimizan el número de violaciones, o en otras palabras, los pesos que maximizan el número de re-identificaciones entre los dos conjuntos de datos.

Utilizando esta notación, tenemos que la siguiente restricción tiene que satisfacerse para todos los pares $i \neq j$.

$$\mathbb{C}(a_i, b_j) - \mathbb{C}(a_i, b_i) + CK_i > 0.$$

Como $K_i = \{0, 1\}$, se pude usar C como una constante, la cual expresa la *mínima distancia* requerida entre el enlace correcto y el resto. Cuanto más grande es el valor, más enlaces correctos se distinguen de los incorrectos.

Utilizando estas restricciones anteriores y el operador de agregación presentado en la Definición 1, d^2WM_p, se puede definir el siguiente problema de optimización:

$$\text{Minimize: } \sum_{i=1}^{N} K_i \tag{3}$$

Subject to:

$$d^2WM_p(a_i, b_j) - d^2WM_p(a_i, b_i) + CK_i > 0, \ \forall i, j = 1, \ldots, N, i \neq j \tag{4}$$

$$K_i \in \{0, 1\} \tag{5}$$

$$\sum_{i=1}^{n} = 1 \tag{6}$$

$$p_i \geq 1 \tag{7}$$

Como se puede observar, este es un problema de optimización con una función objetivo (Ecuación (4)) y unas restricciones (Ecuación (5)) lineales. Debido al operador de agregación usado, se han tenido que añadir un par de restricciones al problema. Las Ecuaciones (6) y (7) hacen referencia a las restricciones introducidas por el uso de pesos de la media ponderada.

Teniendo en cuenta que N es el número de registros y n el número de variables de los dos conjuntos de datos X y Y, se puede calcular fácilmente el número total de restricciones del problema. N^2 restricciones de la Ecuación (4), N son las restricciones necesarias para la Ecuación (5), 1 restricción para la suma de todos los pesos, Ecuación (6), y finalmente n restricciones de la Ecuación (7). Por lo tanto el problema tiene un total de $N^2 + N + n + 1$ restricciones.

IV. Análisis experimental

El método presentado en la sección anterior ha sido evaluado utilizando diferentes conjuntos de datos protegidos. Para la protección de datos se ha utilizado la *Microaggregation* [5], un método muy conocido para la protección de microdatos. Este método proporciona privacidad mediante la agrupación de los datos en pequeños grupos de k elementos, y posteriormente reemplazando los datos originales de cada agrupación por su correspondiente centroide. El parámetro k determina el grado de protección aplicado: cuanto mayor es el valor de k, mayor es la protección aplicada y a su vez mayor es la información perdida en el proceso.

Se han considerando los conjuntos de datos con los siguientes parámetros de protección:

- *M4-33*: 4 atributos microaggregados en grupos de 2 con $k = 3$.
- *M4-28*: 4 atributos, los primeros 2 atributos con $k = 2$, y los últimos 2 con $k = 8$.
- *M4-82*: 4 atributos, los primeros 2 atributos con $k = 8$, y los últimos 2 con $k = 2$.
- *M5-38*: 5 atributos, los primeros 3 atributos con $k = 3$, y los últimos 2 con $k = 8$.
- *M6-385*: 6 atributos, los primeros 2 atributos con $k = 3$, los siguientes 2 atributos con $k = 8$, y los últimos 2 con $k = 5$.
- *M6-853*: 6 atributos, los primeros 2 atributos con $k = 8$, los siguientes 2 atributos con $k = 5$, y los últimos 2 con $k = 3$.

Para cada uno de los conjuntos de datos se han protegido 400 registros aleatoriamente extraídos del censo [19] del proyecto *European CASC* [20], el cual contiene 1080 registros y 13 atributos, y ha sido extensamente usado en otras investigaciones como [21], [22], [23]. Los grados de protección, k, aplicados, varían entre la mínima protección posible, $k = 2$ y un buen grado de protección, $k = 8$, según [13].

	d^2AM	d^2MD	d^2WM
M4-33	0,84	0,94	0,955
M4-28	0,685	0,9	0,93
M4-82	0,71	0,9275	0,9425
M5-38	0,3975	0,8825	0,905
M6-385	0,78	0,985	0,9925
M6-853	0,8475	0,98	0,9875

Tabla I
RESULTADOS EN EL ENLACE DE REGISTROS BASADO EN DISTANCIAS.

En la Tabla I se muestran los resultados de aplicar los métodos estándar de enlace de registros basado en distancias, d^2AM y d^2MD, y el nuevo método supervisado basado en la media ponderada, d^2WM, presentado en la Sección III. Los valores de dicha tabla son el ratio de registros correctamente re-identificados, de modo que 1 significa que el 100 % de las re-identificaciones fueron correctas.

Como se puede apreciar, el método presentado obtiene un incremento relevante en el número de re-identificaciones cuando es comparado con los métodos estándar actualmente utilizados. Este incremento es especialmente importante al comparar d^2WM con la distancia Euclídea, en el cual vemos un incremento de hasta un 50 % para el conjunto *M5-38*.

V. CONCLUSIÓN

En este artículo se ha introducido una variante de enlace de registros basado en distancias. Nuestra propuesta utiliza un algoritmo de aprendizaje supervisado el cual gracias a una media ponderada permite determinar los pesos de esta que maximizan el número de re-identificaciones entre el fichero original y el protegido. De este modo, se han mejorado los sistemas estándar de evaluación del riesgo de un fichero protegido.

Este y otros trabajos en la misma línea de investigación, el enlace de registros basado en distancias, se pueden encontrar en los siguientes artículos [24], [25].

AGRADECIMIENTOS

Esta investigación está parcialmente financiada por el MICINN (proyectos ARES-CONSOLIDER INGENIO 2010 CSD2007-00004, TIN2010-15764 y TIN2011-27076-C03-03) y por by the EC (FP7/2007-2013) Data without Boundaries (número de subvención 262608). Algunos de los resultados presentados en este artículo han sido obtenidos gracias al Centro de Supercomputación de Galicia (CESGA). El trabajo contribuido por el primer autor ha sido parte de un programa de doctorado en Informática de la Universidad Autónoma de Barcelona (UAB).

REFERENCIAS

[1] Dunn, H.L. (1946). Record Linkage. American Journal of Public Health 36 (12), pp. 1412–1416.
[2] Newcombe, H. B., J. M. Kennedy, S. J. Axford, and A. P. James (1959) Automatic Linkage of Vital Records, Science, 130, pp. 954–959.
[3] Newcombe, H. B., J. M. Kennedy (1962). Record linkage: making maximum use of the discriminating power of identifying information. Commun. ACM 5, 11, pp. 563-566.
[4] Fellegi, I., Sunter, A., (1969). A Theory for Record Linkage. Journal of the American Statistical Association 64 (328), pp. 1183¿ –1210.
[5] Defays, D., Nanopoulos, P. (1993) Panels of enterprises and confidentiality: The small aggregates method, Proc. of the 1992 Symposium on Design and Analysis of Longitudinal Surveys, Statistics Canada, pp. 195–204.
[6] Statistics Canada. (2010). Record linkage at Statistics Canada. http://http://www.statcan.gc.ca/record-enregistrement/index-eng.htm
[7] Winkler, W. E. (2003) Data cleaning methods, Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.
[8] Batini, C., Scannapieco, M. (2006) Data Quality - Concepts, Methodologies and Techniques Series: Data-Centric Systems and Applications.
[9] Elmagarmid, A., Panagiotis G., Verykios, V., (2007). Duplicate Record Detection: A Survey. IEEE Transactions on Knowledge and Data Engineering 19 (1), pp. 1–16.
[10] Agrawal, R., Srikant, R. (2000) Privacy-preserving data mining. ACM Sigmod Record, vol. 29, issue 2, pp. 439-450.
[11] Willenborg, L., De Waal, T. (2001) Elements of Statistical Disclosure Control. Springer Verlag.
[12] Spruill, N.L., (1982) Measures of confidentiality. Proc. Survey Research Section American Statistical Association, pp. 260–265.
[13] Domingo-Ferrer, J., Torra, V., (2001) A quantitative comparison of disclosure control methods for microdata, pp. 111–133 of [15].
[14] Winkler, W.E, (2004) Re-identification methods for masked microdata, Privacy in Statistical Databases 2004, Lecture Notes in Computer Science 3050, pp. 216–230.
[15] Doyle, P., Lane, J., Theeuwes, J., Zayatz, L. (editors) (2001) Confidentiality, disclosure, and data access: theory and practical applications for statistical agencies, Elsevier Science.
[16] Jaro, M. A. (1989) Advances in Record Linkage Methodology as Applied to Matching the 1985 Census of Tampa, Florida. Journal of the American Statistical Society, 84:406, pp. 414–420.
[17] Pagliuca, D., Seri, G., (1999), Some results of individual ranking method on the system of enterprise accounts annual survey, Esprit SDC Project, Delivrable MI-3/D2.
[18] Torra, V., Narukawa, Y., (2007) Modeling decisions: information fusion and aggregation operators, Springer.
[19] U.S. Census Bureau. Data Extraction System, 2011, http://www.census.gov/.
[20] Brand, R., Domingo-Ferrer, J., Mateo-Sanz, J.M., (2002) Reference datasets to test and compare SDC methods for protection of numerical microdata. Technical report, European Project IST-2000-25069 CASC.
[21] Laszlo, M., Mukherjee, S., (2005) Minimum spanning tree partitioning algorithm for microaggregation. IEEE Transactions on Knowledge and Data Engineering 17(7), pp. 902–911.
[22] Domingo-Ferrer, J., Torra, V., (2005) Ordinal, Continuous and Heterogeneous k-Anonymity Through Microaggregation, Data Mining and Knowledge Discovery 11(2), pp. 195–212.
[23] Yancey, W., Winkler, W.E., Creecy, R., (2002) Disclosure risk assessment in perturbative microdata protection. In Inference Control in Statistical Databases, Lecture Notes in Computer Science 2316, pp. 135–152.
[24] Abril, D., Navarro-Arribas, G., Torra, V., (2012). Improving record linkage with supervised learning for disclosure risk assessment. Information Fusion, 13(4), 274–284.
[25] Abril, D., Navarro-Arribas, G., Torra, V., (2012). Choquet integral for record linkage. Annals of Operations Research, 195(1), 97–110.

Gestión de identidades digitales basada en el paradigma de la reducción de tiempo de exposición

Jose María Alonso
Telefonica Digital Identity - Privacy
Security researcher
Email: chema@11paths.com

Antonio Guzmán
Telefonica Digital Identity - Privacy
Security researcher
Email: antonio.guzman@11paths.com

Alfonso Muñoz
Telefonica Digital Identity - Privacy
Security researcher
Email: alfonso.munoz@11paths.com

***Resumen*—La presente investigación analiza la problemática actual de la gestión de identidades digitales centrada en tres pilares fundamentales: la seguridad de la solución, su usabilidad y el coste de su implementación. En este artículo se profundiza en la posibilidad de utilizar el paradigma de la reducción de tiempo de exposición para garantizar una mejor aproximación a la gestión de identidades, dando lugar a una gestión robusta, usable y de menor coste que soluciones previas. La argumentación teórica se justifica con el desarrollo de la infraestructura de gestión de identidades Latch, analizando datos reales en escenarios de comunicación comunes en Internet**

***Palabras clave*—gestión de identidades, tiempo de exposición, Latch**

I. Introducción

Cada individuo que accede al mundo digital acumula múltiples identidades digitales, cada una correspondiente con la forma en la que se decide interactuar con los diferentes servicios. El número de identidades cuya suma converge en una única identidad física se incrementa constantemente, a pesar de los esfuerzos invertidos en el desarrollo y adopción de esquemas de federación que permiten delegar los procesos de autenticación y autorización en terceros de confianza [1] [2] [3]. Algunas de las explicaciones que hay detrás de este comportamiento son:

- La desconfianza en los propietarios de los servicios digitales que usan los usuarios.
- Ver en la generación de identidades digitales la posibilidad de ganar en anonimato.
- Definición de contenedores que permitan parcelar diferentes regiones de nuestras vidas digitales (trabajo, amigos, familia, etc.).
- Evaluación de la tecnología.

En la práctica, si no se es muy escrupuloso en la forma en la que se utilizan estas identidades, aparecerán relaciones entre todas ellas. De hecho, lamentablemente, es una realidad que una mayoría de usuarios repiten las contraseñas en más de un servicio [4]. Esto plantea un escenario en el que, si el usuario ha elegido una contraseña que sea deducible, podrá comprometer la seguridad de todas las identidades que confíen en un esquema login-password. Incluso si la contraseña que el usuario ha elegido es una contraseña fuerte, si esta contraseña se usa en un sistema de seguridad débil, podrá ser capturada por un atacante. De nuevo si el usuario ha utilizado la misma contraseña en más de un servicio la fortaleza de las medidas de seguridad de estos no tendrá ningún valor. Este uso indebido de las contraseñas es tan solo una de las razones por las que las cifras que modelan el robo de identidad no dejan de crecer [5].

Este robo de identidades es uno de los problemas más importantes según el informe[6]. Es un problema que pone manifiesto la ineficacia de las soluciones propuestas para proteger la forma en la que los usuarios acceden a los servicios. Aunque el robo de identidades puede localizarse en diferentes puntos de los sistemas informáticos o estar debido a equivocadas actitudes de los usuarios, es en los mecanismos de autenticación y de autorización donde esta amenaza se convierte en un ataque al permitir que un usuario ilegítimo tenga acceso a recursos sólo accesibles a los usuarios legítimos. Aunque existen multitud de mecanismos para resolver estos procesos de autenticación y autorización todos ellos apuestan por aumentar la complejidad de la contraseña para impedir su suposición o robo [7][8] y algunos apuestan por minimizar la probabilidad de que una contraseña se reutilice para varios servicios[8]. En [9] se propone un criterio de evaluación que permite comparar unas soluciones frente a otras. Este criterio propone tres aspectos a evaluar que a su vez queda descompuesto en diversas métricas: la seguridad que ofrece una solución determinada, la usabilidad y el coste de su implementación. En la aplicación de este criterio no es posible encontrar ninguna solución que maximice los tres aspectos de su definición simultáneamente.

El NIST estadounidense propone la siguiente relación de amenazas definidas sobre los sistemas autenticación y, por extensión, sobre los sistemas de autorización [10].

- Online Guessing.
- Phising.
- Pharming.
- Eavesdropping.
- Replay.
- Session hijack.
- Man-in-the-middle (MitM).
- Denial of Service.
- Malicious code (Man-in-the-device (MitD) or Man-in-the-Browser (MitB))

Aquellas soluciones que más seguridad ofrecen como son los mecanismos de autenticación y autorización basados en el uso de token hardware [10] proponen soluciones de seguridad frente a todas las amenazas listadas anteriormente. Sin embargo, adolecen de una baja usabilidad y su coste de implementación es muy alto. En este artículo se propone una solución de seguridad que propone un nivel de seguridad comparable a estos token hardware pero sin menoscabar su usabilidad y simplificando la complejidad de su adopción por los proveedores de los servicios.

II. Paradigma de la reducción del tiempo de exposición

La mayoría de los modelos de seguridad en los que se basan las soluciones existentes realizan una serie de asunciones para determinar cuál es el escenario en el que se propone su utilización. Es frecuente encontrar asunciones que consideran que los atacantes dispondrán de recursos infinitos para la implementación de sus ataques. Estos recursos son los medios materiales y el tiempo de los que podrán disponer para vulnerar las medidas de seguridad. En base a esta asunción es posible hacer una estimación relativa a qué amenazas se pretende hacer frente cuando se propone una medida de seguridad.

La idea principal de este trabajo se centra en proponer una propuesta que complemente a los sistemas de autorización y autenticación actuales, por tanto el cambio pudiera ser inmediato y no abrupto, centrándose en el paradigma de la reducción del tiempo de exposición. Si se limita los recursos que un atacante puede aplicar para vulnerar las medidas de seguridad (autenticación y autorización) se debería minimizar el robo de identidades. El tiempo de exposición que un sistema de autorización y autenticación está expuesto a un atacante, cuando un usuario legítimo no tiene intención de autenticarse, es crítico. Analíticamente esta propuesta puede razonarse de la siguiente forma:

Si se define la relación entre el éxito (o fracaso) de un ataque a un sistema de autenticación y el tiempo en que este sistema está accesible (tiempo de exposición) como una probabilidad condicionada $p(SuccessfulAttack—exposed)$ es posible cuantificar el riesgo relativo RR como 1:

$$RR = \frac{p(SuccessfulAttack|exposed)}{p(SuccessfulAttack|unexposed)} > 1 \qquad (1)$$

Es decir, asumimos que existe una relación directa entre la probabilidad de éxito de un ataque sobre un sistema y la exposición de este sistema. Nuestra intuición nos lleva a plantear la hipótesis de que esta probabilidad será menor si existe una reducción en esta exposición que si no se adopta ninguna medida que reduzca dicha exposición. Si, intuitivamente, podemos considerar esta hipótesis como válida, del razonamiento anterior se puede deducir la siguiente expresión 2:

$$OR = \frac{\frac{p(SuccessfulAttack|exposed)}{p(FailedAttack|exposed)}}{\frac{p(SuccessfulAttack|unexposed)}{p(FailedAttack|unexposed)}} > 1 \qquad (2)$$

La ecuación 2 refleja el *odd ratio (OR)* que mide la probabilidad condicionada en el comportamiento de dos grupos, que en esta argumentación están formados por aquellos ataques en los que no hay límite en el tiempo en el que los objetivos del ataque son accesibles y los ataques que si encuentran restricciones en la exposición de estos servicios. Si se considera que OR>1 entonces se puede concluir que existe una mayor probabilidad de éxito de ataque si existe un sistema expuesto continuamente. A partir de este punto, puede deducirse el porcentaje de riesgo atribuible (*attributable risk percentage, ARP*) a la reducción de la exposición de los sistemas, ecuación 3, que indica qué porción de ataques exitosos podrían ser evitados (independientemente del sistema de autorización que utilice un proveedor de servicios) si se minimizara la exposición en relación con todos los casos.

$$ARP = \frac{RR - 1}{RR} \qquad (3)$$

Esta expresión 3 permite estimar, conocido RR, si la inversión requerida para habilitar estos procesos encaminados a reducir el tiempo de exposición es aceptable o no, en comparación con el riesgo a sufrir un ataque y el daño que este ataque puede producir (ARP). La experiencia profesional y el conocimiento técnico de las técnicas de ataque a los sistemas protegidos por la reducción de su exposición, confirman la asunción inicial de que el riesgo relativo es mayor que 1. Las ecuaciones anteriores reflejan claramente la utilidad de incorporar estos principios a los sistemas de autenticación/autorización actuales. No obstante todavía quedan una serie de preguntas en el aire que no es posible resolver sin experimentación:

1. Cómo de costoso/complejo sería implantar este concepto en sistemas de autorización actuales. Entendiendo que el sistema, como tal, no se modifica si no que se le proporciona una capa extra que gestione esta característica.

2. Usabilidad. Por definición un sistema de autenticación es un entorno incómodo para un usuario, es algo que se interpone entre él y los servicios que desea consumir lo más rápido y fácil posible. ¿Supone algún inconveniente esta capa extra de seguridad?

3. Mitigación/utilidad. Por desgracia, no es sencillo estimar de manera específica y pormenorizada el impacto de este tipo de mecanismos en el fraude y vulneración de mecanismos de protección actual en Internet. Existen multitud de informes globales (medidas reales pero agregadas) pero es difícil conocer valores reales para evaluar si un mecanismo de seguridad concreto mejora o no la situación actual en la protección de identidades digitales.

En los siguientes apartados, mediante experimentación, se podrá obtener algunas medidas reales de la utilización de este paradigma en escenarios reales y estimar con mayor precisión

su utilidad si fuera adoptada en el mercado.

III. Pestillos digitales. Minimizar el tiempo de exposición con Latch

La aplicación del paradigma de la reducción de tiempo de exposición a sistema de autenticación en escenarios reales se podría aplicar de diferentes formas. Nuestra propuesta se centra en la utilización del concepto de pestillos (latch, en inglés) digitales, este concepto, como se verá posteriormente, introduce una serie de ventajas notorias en términos de seguridad, anonimato, usabilidad y transparencia. El concepto de pestillo digital es sencillo de entender con un símil cotidiano. Cuando una familia está en su casa además de cerrar la cerradura utilizando sus llaves podría instalar un cerrojo/pestillo que podría utilizar para proporcionar mayor seguridad a su puerta. Las ventajas que introduce esto son las siguientes:

1. La seguridad no depende de las llaves que tenga la familia y que un atacante podría haber duplicado o robado una copia.
2. El pestillo es una capa extra. El usuario decide cuándo está activo o no (tiempo de exposición) no interfiriendo en la forma en la que se implementa la seguridad por parte de un fabricante de sistemas de autenticación (en nuestro caso, la puerta o la seguridad de la cerradura de la misma). Como se verá posteriormente el proveedor de servicios podría considerar o no este pestillo, esto da garantía absoluta de que un gestor de un sistema de autenticación tiene el control del sistema independientemente del estado del pestillo digital.
3. El sistema es muy sencillo. En su generalización a servicios digitales facilitaría homogenizar la seguridad de diferentes cuentas digitales (identidades) con uno o pocos pestillos digitales. Esto ofrece un nivel de seguridad comparable al nivel 4 definido por el NIST [10].

El concepto de pestillo digital se ha llevado a la práctica en el desarrollo de la arquitectura Latch. En esencia, la arquitectura propuesta tiene dos fases: el pareado de cuentas y el modo de operación. El pareado de cuentas (1) supone vincular, sin que ello suponga niguna pérdida de privacidad por parte del usuario, una cuenta de un proveedor de servicios con una cuenta de un usuario de Latch. Información detallada puede encontrarse en [11], esta arquitectura puede resumirse de la siguiente forma:

1. El proveedor de servicios que utiliza un sistema de autenticación determinado podrá disponer de una capa extra (pestillo digital). Esta información la recibirá por un canal específico con las protecciones adecuadas (confidencialidad, integridad y autenticidad). Para facilitar la integración de Latch con la arquitectura del proveedor de servicios se proporcionan SDKs en diferentes lenguajes (.net, ruby, .c, python, php, java, dotnetnuke) así como múltiples plugins (drupal6, drupal7, joomla, prestashop, redmine, wordpress, openvpn, ssh, roundcube, squirrelMail) [11].
2. Latch no interfiere en la forma en que un sistema de autenticación/autorización toma sus decisiones, por tanto el sistema podría obviar esta información. No parece razonable que si implementa esta capa ignore el pestillo definido, no obstante se habilita esta característica para que si existiera algún problema a la hora de recibir la información del pestillo (por ejemplo en un entorno de tiempo real limitado a una respuesta antes de 2 milisegundos) pudiera decidir qué hacer con la autorización concreta de un usuario, permitir acceso o no. Tal es la flexibilidad que actualmente ya existen proveedores de cierta relevancia que utilizan Latch: Telefónica, Movistar, Acens, Tuenti, Grupo Cortefiel, Cajamar, Universidad de la Rioja o la Universidad de Salamanca.
3. La gestión de los pestillos digitales se traslada a una aplicación móvil (disponible para android, iphone, windows mobile y firefox os) [12]. Acercando el control de la identidades digitales al usuario. El usuario mediante la aplicación movil podrá vincular uno o más pestillos con la autenticación de servicios/operaciones concretas de un proveedor. Por tanto, el usuario solo tendrá que hacer ON/OFF en sus pestillos y el proveedor de servicios solo necesitará consultar el estado de los mismos antes de proceder a la autenticación.

Figura 1: Proceso de pareado de cuentas con Latch

Una vez que se ha completado el proceso de pareado de cuentas, el usuario está en disposición de poder determinar cuál es el nivel de exposición de los servicios y operaciones proporcionadas por el proveedor con quien ha contratado los servicios (figura 2). Cuando un usuario solicita alguna de estas

operaciones (e.g. el login en el servicio), el proveedor, que habrá integrado en la lógica de sus sistemas las llamadas básicas que aseguran la lógica de la interacción con Latch, solicitará el estado en el que el usuario en cuestión había decido que, en ese instante, se encontrara la operación. El servidor de Latch recuperará el estado de esta operación y se la devolverá al proveedor. Si el estado de esta operación fuera *bloqueado*, el proveedor podría deducir que está ante un intento fraudulento de acceso y actuar en consecuencia.

Con todas estas características queda claro que es posible diseñar una solución basada en el paradigma de la reducción del tiempo de exposición, enmascarando la complejidad de la plataforma, simplificando su uso y minimizando el coste/tiempo tanto a usuarios como a proveedores. Del mismo modo son notorios aspectos de flexibilidad de la propuesta. Algunos de ellos son:

1. Configuración de políticas de gestión de identidades basadas en múltiples parámetros: tiempo, geolocalización, etc.
2. Delegación del acceso a sus cuentas a otros usuarios. Por ejemplo, sería útil para control parental.
3. Monitorización por parte del usuario de accesos basado en robo de identidad. Permite tomar contramedidas frente al posible robo de claves de acceso.
4. One-time use. Los servicios pueden configurarse para que se habiliten cuando el usuario se autentica pero inmediatamente después no permita autenticarse de nuevo por un atacante que tuviera las claves. Esto lleva al extremo la protección basada en mínimo riesgo de exposición.

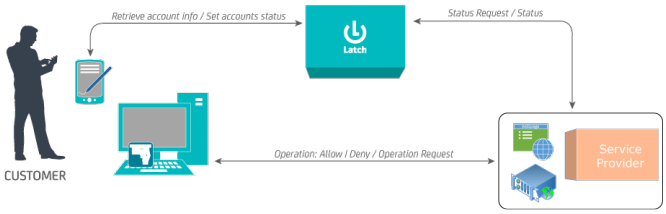


Figura 2: Arquitectura propuesta para garantizar la reducción del tiempo de exposición

En este punto, se puede observar que mediante el diseño y desarrollo de la arquitectura Latch es viable aplicar el paradigma de reducción de tiempo de exposición a entornos reales con sistemas de autenticación variados. Demostrando como estas propuestas pueden adaptarse rápidamente a servicios actuales.

Para concluir nuestra investigación, cubierto ya la investigación analítica y el entorno de experimentación, presenta interés analizar datos reales de cientos de usuarios, a modo de ejemplo, en un CMS (Content Management System) real, así como analizar tendencias en el uso de nuevos mecanismos de seguridad basados en el paradigma que implementa Latch.

IV. Ejemplos de uso

IV-A. Ejemplos de uso en un CMS (CONTENT MANAGEMENT SYSTEM)

En la actualidad la plataforma Latch tiene 4 meses de vida y aunque es un período corto ya cuenta con más de 600 integradores (proveedores de servicios) y miles de usuarios registrados. En la situación actual es posible estudiar una serie de comportamientos relacionados con el uso de los sistemas de autenticación. En esta investigación se centra el foco en el servicio en producción que más usuarios tiene actualmente Latch. Estamos hablando en concreto del Content Management System Joomla [13]. Joomla es un Sistema de gestión de contenidos que permite desarrollar sitios web dinámicos e interactivos. Permite crear, modificar o eliminar contenido de un sitio web de manera sencilla a través de un panel de administración. Latch protege las cuentas de usuarios en su autenticación (independientemente su rol). Joomla tiene pareados 24797 usuarios, lo cual es una cantidad razonable para extraer alguna conclusión real sobre el uso del paradigma de reducción de tiempo de exposición en usuarios reales.

En primer lugar puede observarse como en el 22 % de las solicitudes han intentado accesos ilegítimos (con credenciales válidas). Este acceso ha sido bloqueado dado que el bloqueo estaba activo, el usuario legítimo fue notificado de dicha circunstancia. Del mismo modo se detecta que el 69 % de los usuarios han configurado alguna opción de autobloqueo (temporizador para cerrar cerrojo si abierto), así como existe un número menor de usuarios, un 17 % que desea aprovechar características extras de Latch y utilizarlo como un canal de segundo factor (OTP). Cada vez que alguien se autentique en la web y el cerrojo está abierto se solicitará introducir una clave de un solo uso enviado al móvil del usuario original.

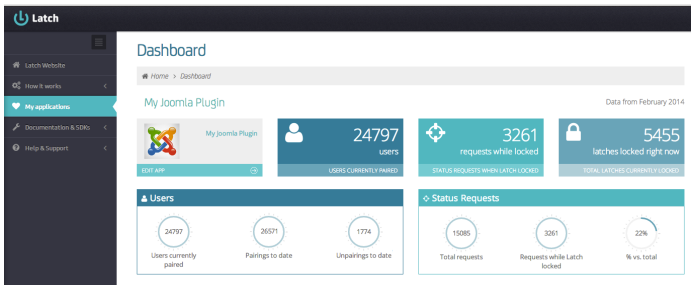


Figura 3: Panel de control simplificado de Latch

IV-B. Ejemplo y tendencias en protección frente a fraude bancario

Es difícil cuantificar el efecto de aplicar una medida de seguridad como es Latch. Aunque si bien es cierto que permite devolver al usuario la sensación de control sobre su vida digital, es complicado trasladar esta sensación a una métrica cuantitativa. Por otro lado, aunque su aplicación sí ofrecería resultados medibles desde la perspectiva de los integradores de esta tecnología, por ejemplo, en términos de prevención del fraude, las políticas de privacidad de las compañías, que

Tabla I: Impacto de Latch en las pérdidas por fraude CNP

	Q4 - 2013	Q1 - 2014	Q2 - 2014	Q3 - 2014	Q4 - 2014	Q1 - 2015	Q2 - 2015	Q3 - 2015	Q4 - 2015
A	0,00012	0,05087	0,1016	0,1524	0,2031	0,2539	0,3046	0,3554	0,4061
B	47779	20804529	41561279	62318029	83074779	103831529	124588279	145345029	166101779
C	$2{,}29{\cdot}10^{8}$	$9{,}99{\cdot}10^{10}$	$1{,}99{\cdot}10^{11}$	$2{,}99{\cdot}10^{11}$	$3{,}99{\cdot}10^{11}$	$4{,}98{\cdot}10^{11}$	$5{,}98{\cdot}10^{11}$	$6{,}98{\cdot}10^{11}$	$7{,}97{\cdot}10^{11}$
D	$2{,}9347{\cdot}10^{-2}$	$2{,}8500{\cdot}10^{-2}$	$2{,}7653{\cdot}10^{-2}$	$2{,}6806{\cdot}10^{-2}$	$2{,}5960{\cdot}10^{-2}$	$2{,}5113{\cdot}10^{-2}$	$2{,}4266{\cdot}10^{-2}$	$2{,}3419{\cdot}10^{-2}$	$2{,}2572{\cdot}10^{-2}$
E	$6{,}73{\cdot}10^{4}$	$2{,}85{\cdot}10^{7}$	$5{,}52{\cdot}10^{7}$	$8{,}02{\cdot}10^{7}$	$1{,}04{\cdot}10^{8}$	$1{,}25{\cdot}10^{8}$	$1{,}45{\cdot}10^{8}$	$1{,}63{\cdot}10^{8}$	$1{,}80{\cdot}10^{8}$
F	43,02	42,60	42,40,48	34,18	43,77	47,55	43,06	47,71	49,32
G	$2{,}90{\cdot}10^{4}$	$1{,}21{\cdot}10^{7}$	$2{,}34{\cdot}10^{7}$	$2{,}74{\cdot}10^{7}$	$4{,}53{\cdot}10^{7}$	$5{,}95{\cdot}10^{7}$	$6{,}25{\cdot}10^{7}$	$7{,}79{\cdot}10^{7}$	$8{,}88{\cdot}10^{7}$

apuestan por el uso de Latch, hacen imposible la publicación de esta información. Por ello, y para finalizar este artículo, se presenta una simulación que pretende estimar a cuánto podría ascender la cantidad prevenida de ser defraudada en caso de que se apostara por Latch como medida de protección de uno de los tipos de fraude que más impacto ha tenido en los últimos tiempos. En [14] se justifica una tendencia descendente en la cantidad defraudada por el uso fraudulento de tarjetas de crédito. A pesar de ello la cantidad absoluta perdida debido a esta lacra en 2013 ha sido próxima 1030 millones de euros. Esta cantidad se divide en tres tipos de fraude que tienen que ver con estas tarjetas: fraude relacionado con los procesos en los que no es posible la comprobación de que efectivamente se está en posesión de la tarjeta (Card Not Present (CNP)) (e.g. procesos de compra por Internet), fraude derivado de su utilización en terminales punto de venta (Point Of Sale (POS)) sin supervisión y derivado de su uso en cajeros automáticos (Automatic Teller Machine (ATM)). Para este estudio concreto, se propone el uso de Latch como medida mitigadora del fraude CNP. Se trata de un escenario en el que la necesidad de comprobar que quién está solicitando una operación es quien dice ser, hoy por hoy, no se ha resuelto eficientemente. Con Latch, y gracias al canal extra de seguridad que facilita, es posible, a día de hoy, demostrar que quién está solicitando la operación, al menos, conoce las credenciales de acceso a Latch.

Para poder estimar en qué medida la implantación de Latch puede suponer un beneficio para, en este caso, las entidades bancarias que emitan tarjetas para sus clientes es necesario modelar la adopción de una nueva tecnología. Para esta labor se han utilizado las métricas propuestas por la International Telecommunication Union [15] para estimar la evolución en la madurez tecnológica de distintas regiones. Esta madurez mide diferentes aspectos de las sociedades tecnológicas, entre los cuáles están el acceso a las nuevas tecnologías y la formación en su uso adecuado. Se ha trasladado la tendencia definida en el informe [16] a la forma en que podría comportarse el número de usuarios de Latch en los países de Europa occidental y se han fijado las condiciones iniciales en el número de usuarios que existen ya para Latch en el primer cuarto de 2014 y el porcentaje de la población de Europa Occidental (409000000 habitantes) que posee un smartphone (56 %). En la fila A de la tabla I, se indica cuál es el porcentaje de adopción esperado de Latch a lo largo del año 2014. En la fila B se traduce este porcentaje al número de usuarios esperados de Latch. Además, en esta adopción, se han considerado una serie de perturbaciones para modelar las fluctuaciones debidas a condicionantes externos (noticias, etc) sobre los gustos de los usuarios. Así, a partir de los usuarios registrados durante el primer cuarto del año 2014 (50000 usuarios de Latch), es posible estimar cuál será el número de usuarios de Latch durante un año. En la fila C se establece la cantidad movida por usuarios de Latch usando sus tarjetas. A partir de este valor, y usando los datos de los informes mencionados, es posible determinar la evolución del fraude en relación con el aumento de transferencias (fila D) y la cantidad defraudada por fraude de tarjetas por usuarios de Latch (fila E). A partir de esta información ha sido posible estimar cuál será la evolución del fraude CNP (fila F) y, por último, el ahorro esperado por el uso de Latch en la fila G.

V. Conclusiones

La presente investigación pone de relieve la problemática actual de la gestión de identidades digitales en Internet. Aunque muchas propuestas se han realizado, entre ellas esfuerzos notorios en esquemas de federación de identidades, hoy día no existe una propuesta definitiva que sin irrumpir bruscamente en cambios a los sistemas de autenticación actuales (y en funcionamiento) proporcione cierta seguridad extra, sea usable y el coste de implementación sea asumible (debe pensarse que en el peor de lo casos se compite con un sistema ampliamente difundido y de bajo coste como es el par usuario-contraseña).

Nuestra propuesta introduce de manera innovadora el concepto de pestillo digital y lleva a la práctica en una plataforma real, Latch, que aunque con pocos meses de vida está siendo utilizada por miles de usuarios. Se analizan datos reales en un escenario de ejemplo y se compara con diferentes estudios globales con el fin de demostrar la viabilidad de aplicar el concepto de reducción del tiempo de exposición a la protección de las identidades digitales.

Referencias

[1] Shibboleth, https://shibboleth.net/

[2] openID Foundation, http://openid.net/

[3] M. Urueña, A. Muñoz, D. Larrabeiti, “Analysis of privacy vulnerabilities in single sign-on mechanisms for multimedia websites”. *Multimedia Tools and Applications 2014*, Volume 68, Issue 1, pp 159-176. Doi: 10.1007/s11042-012-1155-4.

[4] SsplashData, “The 2013 list of worst passwords”, http://splashdata.com/press/worstpasswords2013.htm

[5] Visa, “Visa Europe 2013 Annual Report Enabling new commerce and delivering growth”, http://annualreport.visaeurope.com/downloads/visa_ar2013_complete.pdf

[6] D. Charoen, "Password Security". *International Journal of Security (IJS)*, Volume (8): Issue (1):2014
[7] M. Raza, M. Iqbal, M. Sharif, W. Haider, "A Survey of Password Attacks and Comparative Analysis on Methods for Secure Authentication". *World Applied Sciences Journal* 19 (4): 439-444, 2012. ISSN 1818-4952; DOI: 10.5829/idosi.wasj.2012.19.04.1837
[8] S. Komanduri, R. Shay, P. Gage, ;. Mazurek, L. Bauer, N. Christin, L. Cranor, S. Egelman, "Of Passwords and People: Measuring the Effect of Password-Composition Policies". CHI 2011, May 7-12, 2011, Vancouver, BC, Canada
[9] J. Bonneau, C. Herley, P. Oorshot, F. Stajano, "The quest to replace passwords: A framework for comparative evaluation of web authentication schemes". *In Security and Privacy (SP), 2012 IEEE Symposium* on (pp. 553-567). IEEE.
[10] W. Burr, D. Dodson, E. Netwon, R. Perlner, W. Timothy, S. Gupta, E. Nabbus, "NIST Special Publication 800-63-1. Electronic Authentication Guideline", December 2011. http://csrc.nist.gov/publications/nistpubs/800-63-1/SP-800-63-1.pdf
[11] Latch, "Tu interruptor de seguridad digital". https://latch.elevenpaths.com/www/index.html
[12] ElevenPaths, "Latch. Añade un nivel adicional de protección a tus servicios digitales". http://goo.gl/ksXfm7
[13] Joomla, "Content Management System", http://www.joomla.org/
[14] European central bank, "Second Report on card fraud", july 2013. https://www.ecb.europa.eu/pub/pdf/other/cardfraudreport201307en.pdf
[15] International Telecommunication Union, http://www.itu.int/
[16] Measuring the information Society 2012, http://goo.gl/iW1yWt

Sistema P2P de protección de la privacidad en motores de búsqueda basado en perfiles de usuario

Cristina Romero-Tris, Alexandre Viejo, Jordi Castellà-Roca, Youssef Benkaryouh

Universitat Rovira i Virgili, UNESCO Chair in Data Privacy
Departament d'Enginyeria Informàtica i Matemàtiques
Av. Països Catalans 26, E-43007 Tarragona, Spain
Email: {cristina.romero, alexandre.viejo, jordi.castella}@urv.cat
youssef.benkaryouh@estudiants.urv.cat

***Resumen*—Los motores de búsqueda en Internet (como por ejemplo Google, Bing, Yahoo, AOL, etc.) almacenan en sus servidores las consultas efectuadas por los usuarios. Esta información les permite crear perfiles, y así mejorar el servicio ofrecido (resultados personalizados, sugerencias, correcciones, etc.). Sin embargo, estos perfiles pueden también comprometer el derecho a la privacidad de los usuarios. La información que contienen puede servir para identificar a un usuario y así relacionar su identidad con consultas personales y confidenciales. Por esta razón, es necesario aplicar alguna medida de control que proteja la privacidad de los usuarios de motores de búsqueda. Este artículo presenta un entorno P2P diseñado para permitir que los usuarios se agrupen en diferentes categorías en función de su perfil y puedan ejecutar un protocolo para proteger su privacidad.**

Palabras clave*—Motor de búsqueda *(Web Search Engine)*, Perfil de usuario *(User Profile)*, Privacidad *(Privacy)*, Recuperación privada de la información *(Private information retrieval)*, Servicios personalizados *(Customized Services)*, Sistema P2P *(P2P System)

I. Introducción

Los motores de búsqueda en Internet (en inglés, *Web Search Engines* - WSEs) son una herramienta básica para encontrar contenidos y páginas en Internet. Los motores de búsqueda almacenan información de las páginas Web, la indexan y responden a las consultas de los usuarios con una lista de resultados que corresponden a los enlaces a las páginas que contienen la información buscada.

El éxito de un motor de búsqueda respecto a los otros reside en la adecuación de los resultados a los intereses del usuario. Por ejemplo, consideremos un usuario que consulta el término "Mercurio". Si el usuario está interesado en la astronomía, el motor de búsqueda debería mostrar los resultados relacionados con el planeta. Por el contrario, si sus intereses se centran en la química, los resultados deberían corresponder al elemento químico.

Por esta razón, para un motor de búsqueda es necesario conocer los intereses de sus usuarios. Para ello, basándose principalmente en el historial de consultas, los motores de búsqueda crean un "perfil" de usuario que permite personalizar los resultados de acuerdo con los intereses de cada usuario.

Aunque los perfiles sirven para mejorar la calidad del servicio, también pueden ser una amenaza para la privacidad. El historial de consultas puede contener información personal que permita identificar de forma única a un usuario. Por ejemplo, un usuario que busca su nombre completo, su número de seguridad social, su residencia, ocupación, etc. Además las consultas pueden contener información sensible como problemas de salud, la orientación sexual, política, religión, etc.

El almacenamiento en servidores remotos de esta información puede suponer un riesgo, y por lo tanto necesita ser protegida. Sin embargo, el escándalo de AOL [1] demostró que los usuarios no pueden confiar en la protección ofrecida por los motores de búsqueda. En este caso, se publicaron 20 millones de consultas realizadas por 658.000 usuarios. Algunos de ellos fueron identificados, y su identidad pudo ser relacionada con las consultas que habían realizado, quedando así su privacidad expuesta.

Los motivos anteriores ponen de manifiesto un compromiso entre la privacidad y la calidad del servicio recibido. Por un lado, los usuarios necesitan tomar alguna medida para proteger su privacidad. Por otro lado, los usuarios pueden ser reticentes a utilizar un sistema que les proporcione privacidad pero cuya respuesta sea lenta, o los resultados que les interesen no estén en las primera páginas. Si un usuario ofusca su perfil de manera significativa, obtendrá una buena protección de su privacidad, pero también recibirá un peor servicio, y viceversa.

En este artículo se propone un método que ofrece un compromiso aceptable entre privacidad, utilidad del perfil y tiempo de respuesta.

I-A. Estado del arte

La protección de la privacidad frente a los motores de búsqueda es un tema tratado con anterioridad en distintos trabajos. Una forma de clasificar estos trabajos es de acuerdo con el número de usuarios que participan en el protocolo: existen protocolos *single-party* y *multi-party*. Los protocolos *single-party* permiten que un usuario proteja su privacidad de forma individual. Los protocolos *multi-party* requieren un grupo de usuarios que colabore para proteger su privacidad.

Los protocolos *single-party* se basan en generar consultas falsas [2] o en modificar las consultas que son enviadas a los motores de búsqueda [3]. Sin embargo, algunas propuestas

(p.e. [4], [5]) muestran que las consultas generadas por una máquina no tienen las mismas características sintácticas y semánticas que las consultas humanas. De acuerdo con los trabajos de [4] y [5], es posible detectar consultas automáticas con una probabilidad de error muy baja (alrededor de $0{,}02\,\%$).

Otra opción dentro de los protocolos *single-party* es usar un canal anónimo como por ejemplo *Tor* [6]. No obstante, en este caso el motor de búsqueda no es capaz de generar un perfil del usuario, ofreciendo una peor calidad del servicio (ver [7] para más detalles).

Por otro lado, los protocolos *multi-party* no están afectados por los errores de detección de consultas falsas, y generalmente son más rápidos que los esquemas basados en canales anónimos. En estos protocolos, los usuarios ofuscan su perfil con consultas falsas generadas por otros usuarios. La obtención de estas consultas falsas se realiza mediante la creación de grupos de usuarios, que pueden ser dinámicos [7], o estáticos [8], [9].

Los protocolos con grupos dinámicos utilizan un servidor para agrupar a los usuarios. Este nodo puede ser un cuello de botella, o puede suponer problemas de seguridad, o sufrir ataques de denegación de servicio.

En los protocolos con grupos estáticos, no es necesario un servidor que cree dinámicamente los grupos. Sin embargo, el problema en este caso es que los grupos contienen los mismos usuarios en cada ejecución del protocolo, existiendo la posibilidad de que usuarios deshonestos creen perfiles de los otros miembros del grupo.

I-B. Contribución y organización del artículo

Considerando las ventajas y desventajas de las soluciones con grupos dinámicos y estáticos, en este trabajo se propone una solución híbrida. El sistema presentado en este artículo clasifica a los usuarios de los motores de búsqueda en grupos, donde cada grupo representa una temática o categoría. Un usuario puede pertenecer a varios grupos (si su perfil, generado a partir de su historial de consultas, contiene diversas categorías) durante una "sesión", es decir, un período de tiempo variable en el que el usuario envía consultas al motor de búsqueda de forma regular. Cuando esta sesión finaliza, el usuario se desconecta del sistema, cambiando la topología de los grupos y haciéndola así dinámica.

La principal contribución respecto a trabajos anteriores es la creación y mantenimiento de perfiles dinámicos. El beneficio que obtiene el usuario del sistema es que enviará siempre consultas cuya temática esté ligada a sus intereses. Ésto protege su privacidad ya que ofusca su perfil (no contiene sus consultas reales), pero mantiene sus intereses (permitiendo la obtención de resultados personalizados).

En la Sección I-A se describen brevemente las principales propuestas para proteger la privacidad de los usuarios de los motores de búsqueda. En la Sección II se presenta la arquitectura propuesta y en la Sección III el protocolo de privacidad. La Sección IV presenta los resultados de la simulación del sistemas propuesto. Finalmente, las conclusiones se describen en la Sección V.

II. Arquitectura propuesta

El objetivo principal de nuestra propuesta es construir una arquitectura P2P que permita que los usuarios se agrupen en diferentes categorías en función de sus perfiles. En esta sección se explica en detalle la arquitectura que compone la red Peer-to-Peer de manera que los usuarios puedan agruparse según su perfil en redes no estructuradas. Para ello, a continuación se definen la estructura del perfil considerado y la topología de la red. En esta sección se explica en detalle la arquitectura que compone la red Peer-to-Peer.

II-A. El vector Perfil

Para definir el vector que caracteriza el perfil del usuario, el sistema utiliza las categorías definidas en el Open Directory project (ODP) [10]. ODP es una clasificación ampliamente aceptada de las diferentes categorías que existen de páginas Web. Esta clasificación se realiza a varios niveles. Por ejemplo, la consulta "Michael Jordan" estaría clasificada en ODP bajo las categorías (de más general a más específica) "deportes : baloncesto: profesional: NBA: jugadores". Por cuestiones de simplicidad, llamaremos L al nivel de la categoría, así la categoría "deportes" se encuentra a nivel $L = 1$, la categoría "profesional" a nivel $L = 2$, etc. Asimismo, llamaremos C_L al conjunto de categorías que se encuentran en un nivel, y s al número de categorías que contiene el nivel. Por ejemplo, en el nivel $L = 1$ hay $s = 16$ categorías, $C_1 = \{c_1, \ldots, c_{16}\}$=$\{$arte, negocios, ordenadores, juegos, salud, hogar, infancia y adolescencia, noticias, ocio, referencia, regional, ciencia, compras, sociedad, deportes, mundo$\}$.

Definimos también $Q_i = \{q_1, \ldots, q_k\}$ como el conjunto de consultas generadas por el usuario i, y $E_i^L = \{c_1, \ldots, c_k\}$ como el conjunto de categorías de nivel L a las que pertenecen las consultas de Q_i. Para obtener la categoría a la que pertenece cada consulta, se utiliza el método de análisis textual, como el propuesto en [11]. Este método aplica, para cada consulta, un análisis morfosintáctico y semántico, cuyo resultado final es la categoría ODP a la que pertenece la consulta.

Finalmente, definimos el perfil P_i^L de un usuario i en el nivel L como un vector, donde cada posición corresponde al peso w de una categoría del nivel L, $P_i^L = \{w_{c_1}, \ldots, w_{c_s}\}$. Cada peso w es el número de apariciones de esa categoría en E_i^L.

Por ejemplo, consideremos que nuestro sistema está fijado para $L = 1$. Consideremos también un usuario a que ha generado $k = 3$ consultas $Q_a = \{q_1, q_2, q_3\}$, con categorías E_a^1 =arte, deporte, arte. El perfil resultante sería $P_i^L = \{2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0\}$.

Este perfil es dinámico, es decir, a medida que el usuario vaya enviando más consultas, mayor información contendrá su perfil. Por ejemplo, si el usuario a envía otra query q_4 de arte, y otra q_5 de negocios, su perfil será $P_i^L = \{3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0\}$.

II-B. Estructura de la red

Los usuarios se conectan entre ellos usando una red P2P. Esta red está formada por varios *clusters*, donde cada *cluster*

representa cada una de las categorías de nivel L. Esto quiere decir que, por ejemplo, para $L = 1$ tendremos una red P2P formada por 16 *clusters*.

Dentro de cada *cluster*, existe un nodo llamado *superpeer* que mantiene la lista de direcciones de los nodos que pertenecen a ese *cluster*. Dependiendo de su perfil, un nodo puede estar conectado a uno o varios *clusters*.

Para acceder a la red, se utiliza un servidor *bootstrap*, encargado de mantener y proporcionar la lista de *superpeers* (direcciones IP) a los nuevos nodos.

A continuación se describe con mayor detalle las formas que existen de cambiar la estructura de la red.

II-B1. Nueva conexión de un nodo: Como muchas redes P2P, la red se construye gradualmente a medida que nuevos nodos se conectan al sistema. Los pasos para insertar un nuevo nodo en el sistema son:

1. Consideramos un nuevo usuario i que quiere enviar una consulta q_i. En primer lugar, el sistema extrae la categoría c_j a la que pertenece q_i y crea el perfil inicial del usuario $P_i^L = \{0_{c_1}, \ldots, 1_{c_j}, \ldots, 0_{c_s}\}$
2. El usuario i se conecta al *bootstrap*, que le envía la lista de *superpeers* existentes en el sistema. En este punto, dos situaciones pueden ocurrir:
   - *No hay superpeer para la categoría* c_j. En este caso, i se convierte en el nuevo *superpeer* de esa categoría, y envía un mensaje al servidor *bootstrap*. El *bootstrap* guardará la IP de i en la lista de *superpeers*, asociada con c_j.
   - *Existe un superpeer para la categoría* c_j. El usuario i manda una petición al *superpeer* de c_j para ser incluido en su *cluster*. El *superpeer* le envía la lista de direcciones IP de los nodos que ya pertenecen a c_j, y después incluye la IP de i en esta lista.
     El usuario i utiliza la lista de nodos que le envía el *superpeer* para ejecutar el protocolo de privacidad explicado en la Sección III.

II-B2. Modificación del perfil de un nodo: Hay dos maneras en las que el perfil de un usuario puede ser modificado:

1. *Realiza una nueva consulta de una categoría de su perfil con peso* $w_c > 0$. En este caso, se trata de una categoría ya existente en su perfil, y simplemente se incrementa en una unidad el peso w_c de la categoría.
2. *Realiza una nueva consulta de una categoría de su perfil cuyo peso* $w_c = 0$. En este caso, el perfil cambia el peso de la categoría a $w_c = 1$, y el usuario debe conectarse a un nuevo *cluster*. Utilizando la lista de *superpeers* proporcionada por el *bootstrap*, el usuario vuelve a ejecutar una de las opciones del paso 2.

II-B3. Salida de usuario: Cuando un usuario se desconecta del sistema, la estructura de la red cambia, y los siguientes cambios deben realizarse:

- Si el usuario que se desconecta es un *superpeer*, asumimos que antes de desconectarse envía dos mensajes. El primer mensaje contiene la lista de direcciones IP de los nodos del *cluster*. Este mensaje es enviado a uno de los nodos escogido al azar en el *cluster*, que se convertirá en el nuevo *superpeer*. El segundo mensaje se envía al servidor *bootstrap*, indicando la dirección IP del nuevo *superpeer*.
- Si el usuario que se desconecta es un nodo normal de un *cluster*, simplemente enviará una notificación a su *superpeer* para ser borrado de la lista de nodos del *cluster*. Los nodos que ya estaban en el *cluster* en el momento de la desconexión, borrarán al usuario de sus listas en el caso de que intenten conectarse con él y reciban un mensaje de error indicando que ya no está disponible.

III. Protocolo de Privacidad

La sección anterior explica la estructura de la red P2P. En esta sección, asumimos que cada usuario ya está conectado a la red, y posee la lista de direcciones IP de los nodos de uno o varios *clusters* a los que pertenece su perfil.

En este punto, utiliza un protocolo de privacidad para proteger las consultas que envía al motor de búsqueda. La idea general de este protocolo es que cuando el usuario genera una consulta, no la envía directamente al motor de búsqueda, sino que la envía a otro nodo de la red. Este nodo es escogido al azar dentro del *cluster* al cual corresponde la categoría de la consulta. A su vez, el nodo que recibe la consulta puede aceptarla o rechazarla. Si la rechaza, el usuario buscará a otro nodo del *cluster* que se la acepte. El nodo que acepta la consulta tiene dos posibilidades: enviarla al motor de búsqueda, o reenviarla a otro nodo del *cluster*. El nodo elegirá una opción u otra en base a su historial de consultas enviadas. Si decide reenviar la consulta a otro nodo, éste elegirá de nuevo entre las dos posibilidades, hasta que finalmente un nodo envíe la consulta al motor de búsqueda. Finalmente, los resultados para la consulta se devuelven al nodo que la generó por el camino inverso que siguió la consulta.

A continuación se describen las fases del protocolo anterior con más detalle:

III-A. Inicializaciones

Asumimos que un nodo i de la red tiene dos perfiles:

- *Un perfil real* P_i^L, construido a partir de las categorías de las consultas que genera, tal como se explica en la Sección II-A.
- *Un perfil ofuscado* Φ_i^L, con la misma estructura que el perfil real, pero construido a partir las categorías de las consultas que *envía* al motor de búsqueda.

El objetivo del protocolo de privacidad es hacer que el perfil ofuscado se parezca lo máximo posible al perfil real, ofuscando la información que el motor de búsqueda almacena, pero manteniendo su utilidad. Para controlar el nivell de distorsión del perfil se ha definido el parámetro de ofuscación z. Este parámetro representa la diferencia máxima que puede existir entre el peso de una categoría en el perfil real $w_{c_j^P}$ y el peso de esa misma categoría en el perfil ofuscado $w_{c_j^\Phi}$. Es decir, se tiene que cumplir la siguiente condición: $w_{c_j^P} < w_{c_j^\Phi} + z$.

Además, definimos otro parámetro ξ: la probabilidad de rechazo. Este parámetro indica la probabilidad (entre 0 y 1) que tiene un nodo de rechazar una consulta que recibe de otro usuario.

III-B. Envío de la consulta

Esta fase del protocolo se ejecuta cada vez que un usuario i genera una consulta q_i.

1. i calcula la categoría c_j a la que pertenece q_i.
2. i incrementa una unidad el peso $w_{c_j^P}$ de su perfil real.
3. i escoge al azar un nodo v del *cluster* que corresponde a c_j.
4. i y v ejecutan el siguiente protocolo de compromiso de bit para saber si v debería aceptar o rechazar la consulta. Sin este procedimiento, un usuario egoísta podría rechazar siempre las peticiones recibidas.
   a) i escoge al azar un valor X de longitud suficiente y aplica una función resumen criptográfica segura computacionalmente $H()$, obteniendo $x = H(X)$.
   b) i envía x al nodo v.
   c) v escoge al azar otro valor Y, y aplica la misma función de hash obteniendo $y = H(Y)$.
   d) v envía y al nodo i.
   e) i envía X al nodo v, para que verifique si $x == H(x)$.
   f) v envía Y al nodo i, para que verifique si $y == H(Y)$.
   g) Si alguna de las verificaciones falla, los nodos se desconectan y el usuario i recomienza esta fase del protocolo eligiendo otro v'.
   h) Si las verificaciones se llevan a cabo con éxito, i y v calculan un hash que concatena X e Y: $H(X||Y)$. El resultado de este hash se usa como entrada para un generador pseudo-aleatorio que genera un valor (λ) de manera aleatoria entre 0 y 1. Si $\lambda \geq \xi$, v acepta la consulta q_i. Si $\lambda < \xi$, v rechaza la consulta y el usuario i recomienza esta fase del protocolo eligiendo otro v'.
5. Suponiendo que el nodo que finalmente acepta la consulta es v, éste calcula $w_{c_j^{\Phi_v}} + z$, y lo compara con $w_{c_j^{P_v}}$.
   - Si $w_{c_j^P} \leq w_{c_j^\Phi} + z$, v envía la consulta al motor de búsqueda. Además, incrementa una unidad el peso $w_{c_j^\Phi}$ de su perfil ofuscado.
   - Si $w_{c_j^P} > w_{c_j^\Phi} + z$, v escoge al azar otro nodo r del *cluster* que corresponde a c_j. En este punto, el protocolo vuelve a ejecutarse entre v y r a partir del paso 4.

III-C. Recepción de los resultados

La fase del protocolo anterior se ejecuta hasta que uno de los nodos finalmente envía la consulta al motor de búsqueda. Este nodo es responsable de comenzar el reenvío de los resultados que recibe del motor de búsqueda. Ninguno de los nodos conoce el camino completo que siguió la consulta, simplemente reciben los resultados del nodo al que se la reenviaron, y los pasan al nodo del que la recibieron. Así, finalmente, la consulta llega hasta el usuario que la generó.

IV. Análisis del sistema

Con el objetivo de analizar el sistema propuesto, se ha implementado una aplicación que simula las consultas que enviaría cada usuario al WSE si utilizara este sistema. Además, esta aplicación analiza diversas estadísticas de las simulación, como el número de saltos que realiza una consulta antes de recibir la respuesta. Comparando el perfil original y el perfil obtenido en la simulación, se puede verificar si las categorías se mantienen en el perfil, permitiendo personalizar los resultados de futuras consultas. Otro de los parámetros analizados es el número de saltos de cada consulta, que permite estimar el tiempo de espera. Es decir, si el sistema requiere un gran número de saltos, esto supondría una gran espera y por lo tanto los usuarios serían reticentes a su utilización.

Los datos empleados en la simulación son los proporcionados por AOL. Primero, se han ordenado todas las consultas por la fecha en que fueron realizadas. El simulador recibe cada consulta en su tic correspondiente, es decir, para cada segundo. Los perfiles ofuscados de los usuarios de AOL serían los que hubieran obtenido si hubieran utilizado nuestro sistema.

Dado el gran número de datos de AOL únicamente se ha considerado un día entero para hacer la simulación. El trabajo futuro sería ampliar la simulación a más días.

El simulador y el protocolo incluyen diferentes parámetros que afectan a su comportamiento:

- *Ofuscación* (z): nivel máximo permitido de distorsión del perfil.
- *Probabilidad de rechazo* (ξ): un nodo rechazaría la consulta en función de una probabilidad fijada ξ. Esta probabilidad se obtiene mediante el protocolo de compromiso de bit.
- *Nivel de las categorías (L)*: se ha fijado el primer nivel $L = 1$ al hacer las simulaciones.

El nivel de las categorías es el mismo para todas las simulaciones realizadas, pero el nivel de ofuscación y la probabilidad de rechazo cambian para ofrecer un análisis más completo. Más concretamente, el sistema se ha simulado para tres valores de ofuscación ($z = 0, 5, 10$), y cuatro probabilidades de rechazo ($\xi = 0.25, 0.5, 0.75, 1$).

A continuación, se muestra el promedio de resultados para tres usuarios diferentes en cada una de las configuraciones. La Tabla I muestra el número de saltos promedio que ha necesitado una consulta para ser enviada al motor de búsqueda y recibir los resultados, es decir, ida y vuelta: el número de nodos que han reenviado la consulta y los resultados. Los resultados muestran que el número de saltos se encuentra sobre de los 4 saltos por consulta. También se observa que un nivel de ofuscación de $z = 0$ obliga a realizar un mayor número de saltos hasta encontrar un usuario que quiera enviar la consulta al motor de búsqueda. Esto ocurre porque los usuarios tienen menor flexibilidad para alejarse de su perfil real, restringiendo el número de consultas falsas que pueden enviar. Además, la tabla muestra que, para $z = 0$, la probabilidad de rechazo

ξ afecta ligeramente a los resultados. Cuanto mayor es la probabilidad de rechazo, mayor es el número de saltos. Por otro lado, los resultados no parecen estar afectados por esta probabilidad de rechazo cuando $z = 5$ o $z = 10$. Esto se debe a que los nodos tienen flexibilidad suficiente para alejarse de su perfil real, y pueden aceptar más consultas falsas.

Tabla I
PROMEDIO DE SALTOS DE CADA CONSULTA PARA DISTINTOS VALORES DE z Y ξ

Ofusc. (z) \ Prob. rech. (ξ)	**0,25**	**0,5**	**0,75**	**1**
0	4,32	4,36	4,45	4,62
5	3,48	3,46	3,51	3,42
10	3,43	3,41	3,45	3,41

El siguiente punto a analizar es la relación entre el número de consultas propias que el usuario ha enviado al motor de búsqueda, y el número total de consultas generadas. Por ejemplo, consideremos un usuario que genera 171 consultas, y ejecuta el protocolo propuesto para distribuirlas entre los usuarios de la red. Por distintas razones (e.g., la consulta le ha sido reenviada en un ciclo, o ningún usuario la ha aceptado), al final tiene que enviar él mismo 23 de esas consultas. Entonces, decimos que el motor de búsqueda conoce 23 de las 171 consultas reales generadas por el usuario, y por lo tanto, el nivel de conocimiento que tiene de su perfil es de $23/171 = 0{,}135$. La Tabla II muestra estos valores promedios para cada una de las configuraciones. En estos resultados podemos observar resultados muy similares para todas las configuraciones, independientemente de z y de ξ. Esto quiere decir que estos parámetros afectan al número de saltos que realizará la consulta, pero no a las razones por las que un usuario envía su propia consulta.

Tabla II
PROMEDIO DEL PORCENTAJE DEL PERFIL CONOCIDO POR EL MOTOR DE BÚSQUEDA PARA DISTINTOS VALORES DE z Y ξ

Ofusc. (z) \ Prob. rech. (ξ)	**0,25**	**0,5**	**0,75**	**1**
0	0,30	0,30	0,29	0,24
5	0,28	0,29	0,27	0,30
10	0,29	0,29	0,28	0,30

Por último, las simulaciones realizadas analizan la distancia entre el perfil real del usuario y su perfil ofuscado. Para ello, dado el perfil real $P_i^L = \{w_{c_1^P}, w_{c_2^P}, \ldots, w_{c_s^P}\}$, y el perfil ofuscado $\Phi_i^L = \{w_{c_1^\Phi}, w_{c_2^\Phi}, \ldots, w_{c_s^\Phi}\}$, la distancia entre ambos $d(P, \Phi)$ se calcula aplicando la siguiente fórmula:

$$d(P,\Phi) = \sqrt{(w_{c_1^P} - w_{c_1^\Phi})^2 + (w_{c_2^P} - w_{c_2^\Phi})^2 + \ldots + (w_{c_s^P} - w_{c_s^\Phi})^2}$$

La Tabla III muestra el promedio de distancias entre el perfil real y el ofuscado de cada usuario para cada configuración. De estos resultados podemos extraer que, cuanta más flexibilidad hay en el límite de ofuscación z, mayor será la distancia entre el perfil real del usuario y su perfil ofuscado. Por el contrario, la probabilidad de rechazo ξ no parece afectar de forma regular a los resultados. Esto se debe a que el hecho de aceptar más o menos consultas, no está relacionado con la ofuscación, y por lo tanto con la distanci entre perfiles.

Tabla III
PROMEDIO DE SALTOS DE CADA CONSULTA PARA DISTINTOS VALORES DE z Y ξ

Ofusc. (z) \ Prob. rech. (ξ)	**0,25**	**0,5**	**0,75**	**1**
0	14,46	8,23	9,89	19,53
5	22,34	19,30	24,78	21,21
10	22,94	23,02	27,79	21,54

V. CONCLUSIONES Y TRABAJO FUTURO

Los motores de búsqueda en Internet crean *perfiles* de sus usuarios para personalizar los resultados y ofrecer un mejor servicio. La información almacenada en el perfil puede suponer una amenaza para la privacidad del usuario. Por esta razón, en este trabajo se ha propuesto una arquitectura P2P que permite que los usuarios se agrupen según sus perfiles, permitiendo al mismo tiempo conservar un perfil ofuscado (o sintético) muy similar a su perfil real. Una vez en grupos, los usuarios ejecutan un protocolo con el que envían sus consultas de manera que el motor de búsqueda obtiene un perfil que no se corresponde al del usuario en lo que se refiere a las consultas pero si a las categorías. Esto sirve para favorecer una personalización de los resultados por parte del motor de búsqueda, mientras se protege la privacidad del usuario.

Para analizar el protocolo propuesto, varias simulaciones se han llevado a cabo con distintas configuraciones de parámetros. De estas simulaciones, se han mostrado los resultados que corresponden al número de saltos, al porcentaje del perfil que posee el motor de búsqueda, y a la distancia entre el perfil obtenido tras ejecutar el protocolo (ofuscado) y el perfil real del usuario. Los resultados muestran que (1) un nivel máximo de ofuscación nulo aumenta el número de saltos de las consultas, (2) los porcentajes de perfil que obtiene el motor de búsqueda no están afectados por el nivel de ofuscación ni la probabilidad de rechazo, y (3) la distancia entre el perfil real y el ofuscado está estrechamente relacionada con el nivel máximo de ofuscación z.

Como trabajo futuro, se prevee realizar más simulaciones ajustando otros parámetros del sistema, como el número mínimo de conexiones que debe tener un usuario o el procentaje de usuarios egoístas presentes en la red. Además, el trabajo futuro también incluye una propuesta para controlar el origen de consultas "ilegales", i.e., que un usuario pueda demostrar que no generó una consulta, sino que estaba enviándola para beneficio de otro usuario de la red.

REFERENCIAS

[1] M. Barbaro and T. Zeller, "A face is exposed for aol searcher no. 4417749," New York Times, August 2005.
[2] "TrackMeNot," http://mrl.nyu.edu/dhowe/trackmenot, 2013.
[3] J. Domingo-Ferrer, A. Solanas, and J. Castella-Roca, "h(k)-private information retrieval from privacy-uncooperative queryable databases," *Journal of Online Information Review*, vol. 33, no. 4, pp. 1468–1527, 2009.

[4] R. Chow and P. Golle, "Faking contextual data for fun, profit, and privacy," in *Proceedings of the 8th ACM workshop on Privacy in the electronic society – WPES'09*, 2009, pp. 105–108.
[5] S. T. Peddinti and N. Saxena, "On the privacy of web search based on query obfuscation: a case study of trackmenot," in *Proceedings of the 10th international conference on Privacy enhancing technologies – PETS'10*, 2010, pp. 19–37.
[6] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: the second-generation onion router," in *Proceedings of the 13th conference on USENIX Security Symposium*, 2004, pp. 21–31.
[7] J. Castella-Roca, A. Viejo, and J. Herrera-Joancomarti, "Preserving user's privacy in web search engines," *Computer Communications*, vol. 32, no. 13–14, pp. 1541–1551, 2009.
[8] A. Viejo and J. Castella-Roca, "Using social networks to distort users' profiles generated by web search engines," *Computer Networks*, vol. 54, no. 9, pp. 1343–1357, 2010.
[9] A. Erola, J. Castella-Roca, A. Viejo, and J. M. Mateo-Sanz, "Exploiting social networks to provide privacy in personalized web search," *Journal of Systems and Software*, vol. 84, no. 9, pp. 1734–1745, 2011.
[10] ODP, "Open Directory Project," http://www.dmoz.org/, 2013.
[11] D. Sánchez, J. Castellà-Roca, and A. Viejo, "Knowledge-based scheme to create privacy-preserving but semantically-related queries for web search engines," *Inf. Sci.*, vol. 218, pp. 17–30, Jan. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.ins.2012.06.025

Refinamiento Probabilístico del Ataque de Revelación de Identidades

Alejandra Guadalupe Silva Trujillo, Javier Portela García-Miguel, Luis Javier García Villalba
Grupo de Análisis, Seguridad y Sistemas (GASS), Departamento de Ingeniería del Software e Inteligencia Artificial
Facultad de Informática, Despacho 431, Universidad Complutense de Madrid (UCM)
Calle Profesor José García Santesmases, 9, Ciudad Universitaria, 28040 Madrid, España
Email: {asilva, javiergv}@fdi.ucm.es, jportela@estad.ucm.es

***Resumen*—En la actualidad muy pocas empresas reconocen que se encuentran continuamente en riesgo al estar expuestos a ataques informáticos tanto internos como externos. Más allá de simplemente instalar herramientas de protección contra hackers y células del crimen organizado tales como antivirus y firewalls, deben incluir mecanismos adecuados de seguridad en TI que brinden protección a los ataques que son cada vez más complejos. Existen diversos estudios que muestran que aún cuando se aplique el cifrado de datos en un sistema de comunicación, es posible deducir el comportamiento de los participantes a través de técnicas de análisis de tráfico. En este artículo presentamos un ataque a un sistema de comunicación anónimo basado en el ataque de revelación de identidades. El refinamiento probabilístico presenta una mejora sustancial respecto al ataque previo.**

***Palabras clave*—Análisis de tráfico, ataques estadísticos de revelación, comunicaciones anónimas, privacidad. *(Traffic analysis, statistical disclosure attacks, anonymous communications, privacy)*.**

I. Introducción

Empresas, organizaciones y sociedad generan millones de datos diariamente desde diferentes fuentes tales como: operaciones comerciales y mercantiles, redes sociales, dispositivos móviles, documentos, entre otros. La mayor parte de esta información se almacena en bases de datos altamente sensibles. Se consideran datos sensibles aquellos que puedan revelar aspectos como origen racial o étnico, estado de salud presente y futuro, información genética, creencias religiosas, filosóficas y morales, afiliación sindical, opiniones políticas, preferencia sexual y cualquier otro que pueda utilizarse para generar un daño, llámese robo de identidad, extorsión ó fraude por mencionar algunos.

La seguridad en los *data centers* se ha vuelto una de las grandes prioridades ya que tanto los ladrones de datos y células del crimen organizado buscan insistentemente infiltrarse en el perímetro de defensas a través de complejos ataques con un éxito alarmante, derivando en efectos devastadores. Hoy en día estamos inmersos en una sociedad digital donde podemos organizar un evento y enviar una invitación por Facebook; compartir fotos con amigos por medio de Instagram; escuchar música a través de Spotify; preguntar la ubicación de una calle utilizando Google Maps. La información personal es protegida por medio de la legislación y aunque no en todos los países se aplique efectivamente, en el ámbito de la sociedad digital funciona de manera diferente [1]. Toda la información disponible acerca de una persona puede ser referenciada con otra y dar lugar a prácticas de violación de la intimidad.

Cada persona tiene el derecho de controlar su información personal y proporcionarla a ciertas terceras partes. Desde la década pasada se observa una mayor preocupación por cómo se maneja la información privada de los usuarios en el ámbito gubernamental y de las empresas. Y recientemente, después de la filtración de información de un técnico estadunidense de la CIA al mundo, aumentaron las mesas de diálogo, investigaciones y fundamentalmente se creó toda una polémica en torno a la privacidad de los datos y lo expuesto que estamos a ser objetos de monitorización.

Las organizaciones privadas y públicas, así como las personas deben incluir la protección de la privacidad más allá de los típicos aspectos de integridad confidencialidad y disponibilidad de los datos. Aplicaciones utilizadas para garantizar la protección de la privacidad son por ejemplo los sistemas de resistencia a la censura, espionaje, entre otros; algunos de ellos utilzados para ofrecer seguridad a disidentes o periodistas viviendo en países con regímenes represores. Dentro de la misma rama de tecnologías, también existen mecanismos utilizados para acelerar la transición de cifrado como un servicio, que incluye cifrado basado en hardware con almacenamiento de llaves, esquemas de protección centralizada de datos para aplicaciones, bases de datos, ambientes virtuales de almacenamiento, y controles de acceso basados en roles.

Los ataques en las redes de comunicación son un serio problema en cualquier organización. Las nuevas tecnologías tienen un gran reto al buscar mejorar soluciones de seguridad para centros de datos. Se ha probado que el análisis de tráfico y la topología de una red, no proporcionan suficiente protección en la privacidad de los usuarios aún cuando se apliquen mecanismos de anonimato, ya que a través de información auxiliar, un atacante puede ser capaz de menguar sus propiedades. En el contexto de las redes de comunicación, con el análisis del tráfico se puede deducir información a partir las características observables de los datos que circulan por la red tales como: el tamaño de los paquetes, su origen y destino, tamaño, frecuencia, temporizacion, entre otros.

En este artículo nos enfocamos en mostrar cómo el análisis de tráfico de datos puede comprometer el anonimato de un sistema de comunicación anónima a través de técnicas y métodos que arrojen como resultado los patrones de comunicación de

los elementos que la componen.

La composición del presente artículo es de la siguiente manera, en primer lugar la introducción. En la sección II abordamos el estado del arte. La siguiente sección describe el algoritmo utilizado, haciendo énfasis en el refinamiento probabilístico. En la sección IV presentamos la aplicación del algoritmo. Y finalmente en la sección V mostramos las conclusiones sobre los resultados y trabajos futuros

II. Estado del arte

II-A. Privacidad

La definición de privacidad de acuerdo a [2] es el derecho de un individuo a decidir qué información acerca de él mismo puede ser comunicada a otro y bajo qué circunstancias.

Economistas, sociólogos, historiadores, abogados, ingenieros en sistemas informáticos, por mencionar algunos, han adoptado su propia definición de privacidad, tal como su valor, alcance, prioridad y curso de estudio. Detalles relacionados a los antecedentes, legislación e historia de la privacidad se muestran en [3]. De acuerdo a los expertos, privacidad e intimidad son conceptos difíciles de definir; consideramos parte de ello: las condiciones de salud, identidad, orientación sexual, comunicaciones personales, preferencias religiosas, estados financieros, además de muchas otras características. Trabajos relacionados en cómo las PETs se han aplicado desde áreas del entorno económico, social y técnico [4].

Las bases de la legislación respecto a la privacidad datan del año 1948, en la Declaración Universal de Derechos Humanos donde se estableció que ninguna persona debía ser sujeta a interferencias arbitrarias en su privacidad, familia, hogar o correspondencia, así como a su honor y reputación. Pero, a pesar de los avances políticos y legales que se han dado, no ha sido posible resolver algunos de los problemas fundamentales para evitar los abusos que se dan todos los días. La falta de claridad y precisión en los derechos a la libertad de expresión y los límites de información son un problema latente.

El desarrollo e los medios de comunicación digital, el auge de las redes sociales, la facilidad de acceso a dispositivos tecnológicos, está permeando la tranquilidad de miles de personas en su vida pública y privada. Ejemplos abundan, como el caso de una funcionaria de una localidad belga, quien fue sorprendida y videograbada mientras mantenía relaciones sexuales en las oficinas del Ayuntamiento. La grabación fue realizada y subida a Internet por un grupo de jóvenes. Otro escándalo se dio cuando el presidente del Instituto de Seguridad Social de Guatemala quién fue filmado en su oficina cuando realizaba actos poco legales. A diferencia del primer caso, en éste último sí existía un crimen que perseguir y la acción se justificaba para dar a conocer los hechos públicamente.

Como éstos, muchos más casos son parte del material disponible en internet y en los medios convencionales, como los videos que se filtraron de la Viceministra de Cultura y Juventud de Costa Rica, y del concejal del PSOE en Yébenes, España. A nadie parece importar los efectos que continúan afectando vidas, donde la indiferencia parece ser la constante. La participación de los derechos humanos nacionales e internacionales, el gobierno, los medios de comunicación así como la sociedad parecen estar lejanos de este problema. El escándalo a expensas de la intrusión y diseminación de la vida privada e íntima de las personas es inaceptable. Es un círculo vicioso que tiene su origen en la violación de un derecho, pero más cuando se lleva a las redes sociales y de ahí a la mayoría de los medios de comunicación con el pretexto de ser noticia.

II-B. Privacy Enhancing Technologies

La Comisión Europea define las Tecnologías que mejoran la privacidad [5] como “El uso de los PETs puede ayudar a diseñar sistemas de comunicación y servicios de forma que minimiza la recolección y uso de datos personales y facilita el cumplimiento con la regulación de protección de datos”. No hay una definición aceptada por completo de las PETs, así como tampoco existe una clasificación. La literatura relacionada a las categorías de los PETs de acuerdo a sus principales funciones, administración de privacidad y herramientas de protección de privacidad [6] [7] [8]. En general las PETs son observadas como tecnologías que se enfocan en:

a. Reducir el riesgo de romper principios de privacidad y cumplimiento legal.
b. Reducir al mínimo la cantidad de datos que se tienen sobre los individuos.
c. Permitir a los individuos a mantener siempre el control de su información.

Varios investigadores se han centrado en proteger la privacidad y los datos personales por medio de técnicas criptográficas. Las aplicaciones PETs tales como seguros digitales individuales o administradores virtuales de identidad se han desarrollado para plataformas confiables de cómputo. Tradicionalmente las PETs han estado limitadas para proporcionar pseudononimato [9]. En contraste a los datos totalmente anónimos, el pseudononimato permite que datos futuros o adicionales sean relacionados a datos actuales. Este tipo de herramientas son programas que permiten a individuos negar su verdadera identidad desde sistemas electrónicos que operan dicha información y sólo la revelan cuando sea absolutamente necesario. Ejemplos incluyen: navegadores web anónimos, servicios email y dinero electrónico. Para dar un mejor enfoque acerca de las PETs, consideremos la taxonomía de Solove [10] utilizada para categorizar la variedad de actividades que afectan la privacidad. Para mayor información respecto a las propiedades de privacidad en escenarios de comunicación anónimos vea [9].

- Recolección de información: Vigilancia, Interrogatorio.
- Procesamiento de la Información: Agregación, Identificación, Inseguridad, Uso secundario, Exclusión.
- Difusión de la Información: Violación de la confidencialidad, Divulgación, Exposición, Aumento de la accesibilidad, Chantaje, Apropiación, Distorsión.
- Invasión: Intrusiones, Interferencia en la toma de decisiones.

La recolección de la información puede ser una actividad dañina, aunque no toda la información es sensible, ciertos

datos definitivamente lo son. Cuando la información es manipulada, utilizada, combinada y almacenada. se etiqueta a dichas actividades como Procesamiento de la información; cuando la información es liberada, encaja en las actividades conocidas como Difusión de la información. Finalmente, el último grupo de las actividades es la Invasión que incluye violaciones directamente a individuos. Todas estas actividades son parte de las prácticas comunes de las compañías que se dedican a recolectar información, como la preferencia de compras, hábitos, nivel educativo, entre otros. Todo ello por medio de múltiples fuentes para propósitos de venta.

En otras sub-disciplinas de las ciencias computacionales, la privacidad también ha sido motivo de investigación principalmente en como las soluciones de privacidad se pueden aplicar en contextos específicos. En otras palabras, definir el proceso de cuándo y cómo deben aplicarse las soluciones de privacidad. Antes de elegir una tecnología de la protección de privacidad surgen varias preguntas que deben responderse dado que no existe la certeza de que una tecnología soluciona un problema en específico. Una de las preguntas a considerar es quién define qué es la privacidad, el diseñador de tecnologías, los lineamientos de la organización, o los usuarios [11].

II-C. Comunicaciones anónimas

Las comunicaciones anónimas tienen como objetivo ocultar las relaciones en la comunicación. Dado que el anonimato es el estado de ausencia de identidad, las comunicaciones anónimas se pueden lograr removiendo todas las características identificables de una red anónima. Consideremos a un sistema donde se concentra un conjunto de actores en una red de comunicación, tales como clientes, servidor y nodos. Estos actores intercambian mensajes por medio de canales públicos de comunicación. Pitfzmann y Hansen [9] definieron el anonimato como el estado de ser no identificable dentro de un conjunto de sujetos, conocido como el conjunto anónimo. Una de las principales características del conjunto anónimo es su variación en el tiempo. La probabilidad que un atacante puede efectivamente revelar quién es el receptor de un mensaje es exactamente de $1/n$, siendo n el número de miembros en el conjunto anónimo. La investigación en esta área se enfoca en desarrollar, analizar y llevar a cabo ataques de redes de comunicación anónimas. La infraestructura del Internet fue inicialmente planteado para ser un canal anónimo, pero ahora sabemos que cualquiera puede espiar la red. Los atacantes tienen diferentes perfiles tales como su área de acción, rango de usuarios, heterogeneidad, distribución y localización. Un atacante externo puede identificar patrones de tráfico para deducir quiénes se comunican, cuándo y con qué frecuencia.

En la literatura se ha clasificado a los sistemas de comunicación anónima en dos categorías: sistemas de alta latencia y baja latencia. Las primeras tienen como objetivo proporcionar un fuerte nivel de anonimato pero son aplicables a sistemas con actividad limitada que no demandan atención rápida tal como el correo electrónico. Por otro lado, los sistemas de baja latencia ofrecen mejor ejecución y son utilizados en sistemas de tiempo real, como por ejemplo aplicaciones web, mensajería instantánea entre otros. Ambos tipos de sistemas se basan en la propuesta de Chaum [12], quién introdujo el concepto de *mix*. El objetivo de una red de *mixes* es ocultar la correspondencia entre elementos de entrada con los de salida, es decir encubrir quien se comunica con quien. Una red de *mixes* reúne un cierto número de paquetes de usuarios diferentes llamado el conjunto anónimo, y entonces a través de operaciones criptográficas cambia la apariencia de los paquetes de entrada, por lo que resulta complicado para el atacante conocer quiénes se comunican. Los *mixes* son el bloque base para construir todos los sistemas de comunicación de alta latencia [12]. Por otro lado en los últimos años, se han desarrollado también sistemas de baja latencia, como por ejemplo: Crowds [13], Hordes [14], Babel [15], AN.ON [16], Onion routing [17], Freedom [18] and Tor [19]. Actualmente, la red de comunicación anónima más utilizado es Tor, que permite navegar de manera anónima en la web. En [20] se muestra un comparativo de la ejecución de sistemas de comunicación de alta y baja latencia.

II-D. Redes mixes

En 1981, Chaum introduce el concepto de las redes *mixes* cuyo propósito es ocultar la correspondencia entre elementos de entrada con los de salida. Una red de mixes recolecta un número de paquetes desde diferentes usuarios llamado el conjunto anónimo, y entonces cambia la apariencia de los paquetes de entrada a través de operaciones criptrográficas. Lo anterior hace imposible relacionar entradas y salidas. Las propiedades de anonimato serán más fuertes en tanto el conjunto anónimo sea mayor. Un *mix* es un agente intermediario que oculta la apariencia de un mensaje, incluyendo su longitud. Por ejemplo, supongamos que Alice genera un mensaje para Bob con una longitud constante. Un protocolo emisor ejecuta varias operaciones criptográficas a través de las llaves públicas de Bob. Después, la red *mix* oculta la apariencia del mensaje al decodificarlo con la llave privada del *mix*.

El proceso inicial para que Alice envíe un mensaje a Bob utilizando un sistema de *mixes* es preparar el mensaje. La primera fase es elegir la ruta de transmisión del mensaje; dicha ruta debe tener un orden específico para enviar iterativamente antes de que el mensaje llegue a su destino final. La siguiente fase es utilizar las llaves públicas de los *mixes* elegidos para cifrar el mensaje, en el orden inverso en que fueron elegidos. En otras palabras la llave pública del último *mix* cifra inicialmente el mensaje, después el penúltimo y finalmente la llave pública del primer *mix* es usada. Cada vez que se cifra el mensaje una capa se construye y la dirección del siguiente nodo es incluida. De esta manera cuando el primer *mix* obtiene un mensaje preparado, dicho mensaje será descifrado a través de la llave privada correspondiente y será direccionado al siguiente nodo.

Los ataques externos se ejecutan desde fuera de la red, mientras que los internos son desde nodos comprometidos los cuales son de hecho parte de la misma red. Las redes de *mixes* son una herramienta poderosa para mitigar los ataques externos al cifrar la ruta emisor- receptor. Los nodos

participantes de una red *mix* transmiten y retardan los mensajes con el fin de ocultar su ruta. Pero es posible que puedan estar comprometidos y llevar a cabo ataques internos. Este tipo de problema se trata en [13] al ocultar el emisor o receptor de los nodos de transmisión.

II-E. Análisis de tráfico

El análisis de tráfico pertenece a la familia de técnicas utilizada para deducir información de los patrones de un sistema de comunicación. Se ha demostrado que el cifrado por sí mismo no garantiza el anonimato. Aún cuando el contenido de las comunicaciones sean cifradas, la información de enrutamiento debe enviarse claramente ya que los ruteadores deben determinar el siguiente punto de la red a dónde se direccionará el paquete. En [21] se muestran algunos de las técnicas de análisis de tráfico utilizadas para revelar las identidades en una red de comunicación anónima.

II-F. Ataques estadísticos

La familia de ataques estadísticos fue iniciada por Danezis en [22] donde introdujo el ataque estadístico de revelación (*Statistical Disclosure Attack, SDA*). En dicho trabajo se nota que llevando a cabo un amplio número de observaciones por cierto período de tiempo en una red de *mixes*, se puede calcular la probabilidad de distribuciones de envío/recepción de mensajes y con ello menguar la identidad de los participantes en un sistema de comunicación anónimo. A partir de éste ataque se desarrollaron muchos más tomando como base el análisis de tráfico para deducir cierta información a partir de los patrones de comportamiento en un sistema de comunicación.

Los ataques contra redes de *mixes* son conocidos también como ataques de intersección [23]. Se toma en cuenta la secuencia de un mensaje a través de una misma ruta en la red, esto quiere decir que se analiza el tráfico. El conjunto de los receptores más probables se calcula para cada mensaje en la secuencia e intersección de los conjuntos lo que permite conocer quién es el receptor de un determinado mensaje. Los ataques de intersección se diseñan basándose en la correlación de los tiempos donde emisores y receptores se encuentran activos. Al observar los elementos que reciben paquetes durante las rondas en las que Alice está enviando un mensaje, el atacante puede crear un conjunto de receptores más frecuentes de Alice. La información proporcionada a los atacantes es una serie de vectores representando los conjuntos de anonimato observados de acuerdo a los t mensajes enviados por Alice. Dentro de la familia de ataques estadísticos, cada uno de ellos se modela con un escenario muy específico; y en algunos casos poco semejantes al comportamiento de un sistema de comunicación real. Algunos asumen que Alice tiene exactamente m receptores y que envía mensajes a cada uno de ellos con la misma probabilidad, o bien son ataques que se enfocan en un solo usuario como soluciones individuales que son interdependientes, cuando la realidad indica cuestiones diferentes.

III. ALGORITMO

El objetivo de nuestro algoritmo es extraer información relevante sobre las relaciones entre cada par de usuarios. En [24] se describe el problema, así como el marco base y supuestos. Las tablas de las rondas donde se muestran los patrones de comunicación entre usuarios se representan con valores de 1 si existe relación y 0 en caso contrario. El atacante es capaz de observar cuántos mensajes son enviados y recibidos, es decir las sumas marginales por fila y columna de cada ronda 1, ...,T donde T es el número total de rondas. En cada ronda sólo consideramos usuarios que reciben y envían mensajes. Por lo tanto, decimos que un elemento (i, j) está presente en una ronda si las marginales correspondientes son diferentes a 0.

Hemos adoptado el término "cero trivial", que son los elementos que representan pares de usuarios que nunca han coincidido en ninguna ronda, Denotando n_{ij} el contenido del elemento (i, j), n_{i+} el valor marginal de la fila i, n_{+j} el valor marginal de la columna j, n la suma de los elementos y r el número de filas.

Algoritmo 1: Descripción del algoritmo

1. Generar n_{11} de una distribución uniforme entera donde $i = 1, j = 1$;
2. Iniciar un recorrido por columnas, para cada elemento n_{k1} en esta columna hasta $k-1$, se calculan nuevas cotas para n_{k1} a partir de la siguiente ecuación:

$$\text{máx}\big((0, (n_{+1} - \sum_{i=1}^{k-1} n_{i1}) - \sum_{i=k+1}^{r} n_{i+}\big) \leq$$

$$n_{ij} \leq \text{mín}(n_{k+}, n_{+j} - \sum_{i=1}^{k-1} n_{i1})$$

   n_{k1} se genera según un entero uniforme;
3. El último elemento de la fila se rellena automáticamente al coincidir las cotas superior e inferior coinciden, haciendo $n_{(k+1)+} = 0$ por conveniencia;
4. Cuando se completa la columna ésta se elimina de la tabla y se recalculan las marginales por fila n_{i+} y el valor n;
5. La tabla tiene ahora una columna menos y se repite el proceso hasta llenar todos los elementos;

Al final lo que obtenemos son una serie de tablas factibles generadas para cada ronda. Por lo que la media de cada elemento sobre todas las tablas para todas las rondas es una estimación de su valor real. La media obtenida por elemento y ronda se agrega sobre todas las rondas la cual representa un estimado de la tabla agregada $\widehat{A}$. Para cada elemento, se estima la probabilidad de cero, calculando el porcentaje de tablas con elemento cero para cada ronda en que el elemento está presente y multiplicando las probabilidades obtenidas para todas esas rondas. En la tabla resultante los elementos se ordenan por

su probabilidad de cero a excepción de los elementos que son cero triviales. De esta manera, los elementos con menor probabilidad de ser cero son los que se consideran candidatos a tener una relación. Para llevar a cabo la clasificación seleccionamos un punto de corte p y consideramos "celdas cero" si probabilidad de cero $> p$, en tanto las "celdas positivas" son aquellas donde la probabilidad de cero$< 1 - p$. Aquellas celdas que no entran en estas dos categorías se les llama "no clasificadas".

El algoritmo utilizado en [24] presupone inicialmente equiprobabilidad de las tablas extraídas. Al desarrollarlo se obtienen, al margen de una primera clasificación de las celdas (*i*, *j*) en 1 ó 0 según exista comunicación o no entre ese par de usuarios, estimaciones para la tasa de mensajes enviados por ronda para cada celda. A partir de estas estimaciones iniciales, puede volver a desarrollarse el algoritmo en un segundo ciclo, en el cual las tablas no se generan con equiprobabilidad. En el primer ciclo del algoritmo el valor de cada celda en cada tabla-ronda era generado según una distribución uniforme manteniendo las restricciones dadas por la información marginal conocida. En este segundo ciclo existen varias posibilidades teniendo en cuenta las primeras estimaciones:

a. Generar el valor de cada celda en cada tabla-ronda según una distribución de Poisson cuyo parámetro lambda es la tasa estimada de mensajes por ronda para esa celda.
b. Generar el valor de cada celda en cada tabla-ronda según la distribución de probabilidad discreta del número de mensajes por ronda en esa celda. Esta distribución es construida a partir de los resultados del primer ciclo del algoritmo, estimando probabilidades de 0, 1, 2, ... mensajes según su porcentaje relativo de ocurrencias.

Este segundo ciclo puede volver a servir de base para ciclos sucesivos en un proceso iterativo. En los resultados siguientes se ha utilizado la opción b). Para llevar a cabo nuestro ataque, primero por cuestiones pedagógicas, simulamos los datos de un sistema de correo electrónico. Para la generación de rondas definimos el número de usuarios participantes *N*, lambda que es el promedio de mensajes enviados por ronda en la celda (*i*, *j*) y el número de rondas *NR* que se desea generar.

1. Con las rondas simuladas se ejecuta el Algoritmo 1 y se obtienen las tablas factibles de cada ronda. Posteriormente se lleva a cabo un test de clasificación binaria para los elementos calculados, donde 0 en la celda (*i*, *j*) significa que no existe relación entre el emisor *i* y el receptor *j*, en tanto 1 significa que sí hay comunicación entre ellos.
2. Generar métricas características para los tests de clasificación binaria (sensibilidad, especificidad, valor predictivo negativo, valor predictivo positivo).
3. Con la información de las tablas factibles para cada ronda se calculan las frecuencias relativas de 0, 1, 2, ... mensajes para cada celda y se obtiene una aproximación a la distribución de probabilidad del número de mensajes por ronda, a partir de la normalización de esas frecuencias relativas.
4. Se vuelve a ejecutar el algoritmo utilizando las probabilidades estimadas para cada celda, normalizadas en cada caso a sus restricciones, en lugar de la distribución uniforme.
5. Se generan métricas de clasificación binaria y se vuelven a estimar las probabilidades.
6. Se itera el proceso a partir del punto 4.

IV. Aplicación del Algoritmo

Llevamos a cabo un gran número de simulaciones luego de generar rondas. El algoritmo no proporciona soluciones uniformes, dado que algunas tablas son más probables que otras debido al orden utilizado al ir llenando filas y columnas. No nos enfocamos en encontrar soluciones para un solo usuario, por lo que: i) Reordenamos aleatoriamente filas y columnas antes de calcular tablas factibles; ii) Conservamos solo las tablas factibles diferentes.

La Tabla I presenta los resultados obtenidos aplicando los algoritmos anteriormente descritos. Los resultados de la iteración 1 corresponden a la aplicación de lo que llamamos primer ciclo [24]; a partir de la iteración 2 se ejecuta el segundo ciclo y de acuerdo a los resultados que obtuvimos pudimos observar que tres iteraciones nos proporcionaban mejores resultados en la mayoría de los casos. Se puede observar también que la complejidad de las rondas crece cuando el número de usuarios y el número de rondas es mayor.

Tabla I
Resultados de la Simulación

No. de usuarios	Iteración	Sensibilidad	Especificidad	VPP	VPN	% de clasificación
10	1	0.9876	0.5789	0.9166	0.9090	0.91
	2	0.9876	0.9473	0.9473	0.9876	0.98
	3	0.9876	0.9473	0.9473	0.9876	0.98
	4	0.9473	0.9876	0.9473	0.9876	0.98
15	1	0.3225	0.9948	0.9090	0.9018	0.90
	2	0.6774	0.9948	0.9545	0.9507	0.95
	3	0.8387	0.9948	0.9629	0.9747	0.97
	4	0.8064	0.9948	0.9615	0.9698	0.96
20	1	0.1818	0.9857	0.6666	0.8846	0.87
	2	0.7272	0.9857	0.8888	0.9583	0.95
	3	0.8181	0.9857	0.9	0.9718	0.96
	4	0.7272	0.9857	0.8888	0.9583	0.95
25	1	0.2297	0.9969	90.9444	0.8507	0.85
	2	0.4324	1	1	0.8858	0.89
	3	0.5540	1	1	0.9080	0.91
	4	0.6486	1	1	0.9261	0.93
30	1	0.1058	0.9981	0.90	0.8764	0.8768
	2	0.2235	0.9981	0.95	0.8909	0.8928
	3	0.3764	0.9981	0.96	0.9104	0.9136
	4	0.3058	0.9981	0.96	0.9013	0.904
35	1	0.0441	0.9986	0.8571	0.8544	0.85
	2	0.2205	0.9986	0.9677	0.8780	0.88
	3	0.2720	0.9986	0.9736	0.8851	0.89
	4	0.2941	0.9986	0.9756	0.8882	0.89

En la Figura 1 se modela la tasa de clasificación respecto a las veces que se ha iterado el algoritmo. Se puede observar una mejora en el porcentaje de clasificación en todos los casos, en relación a la iteración 1.

V. Conclusiones

En las redes de comunicación, los *mixes* ofrecen protección contra observadores al ocultar la apariencia de los mensajes,

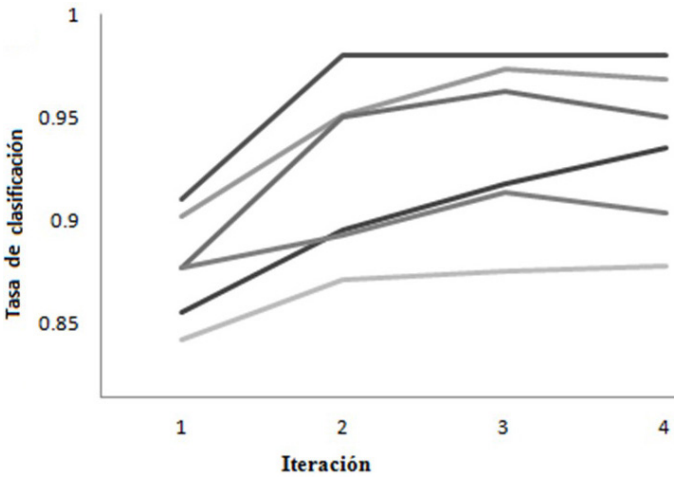


Figura 1. Tasa de clasificación vs. Número de iteración

sus patrones, longitud y enlaces entre emisores y receptores. El objetivo de este trabajo es desarrollar un ataque estadístico global para revelar la identidad de emisores y receptores en una red de comunicaciones que está protegida por técnicas estándar basadas en *mixes*. Para efecto de refinar nuestro ataque tomamos en cuenta las tablas factibles no repetidas, calculamos las frecuencias relativas para cada celda y obtuvimos una aproximación a la distribución de probabilidad del número de mensajes. El método puede ser aplicado en otro tipo de sistemas de comunicación como por ejemplo en redes sociales y protocolos punto a punto; asimismo puede ser implementado fuera del dominio de las comunicaciones como la revelación estadística de tablas públicas y la investigación forense. Nuestro método es afectado por muchos factores como el número de usuarios y el número promedio de mensajes por ronda lo que deriva a una alta complejidad de las tablas que influye de manera negativa en el ataque. El alcance en la tasa de clasificación muestra que entre mayor es el número de rondas se obtienen mejores resultados. Finalmente iteramos el algoritmo. Es necesaria mayor investigación para definir con cuántas iteraciones se pueden ver mejores resultados. De acuerdo a la literatura revisada, podemos concluir que los protocolos de anonimización propuestos hasta ahora consideran escenarios muy específicos. Los ataques estadísticos de intersección se centran en un usuario solamente, sin considerar las relaciones entre todos los usuarios.

Agradecimientos

El Grupo de Investigación GASS agradece la infraestructura proporcionada por el Campus de Excelencia Internacional (CEI) Campus Moncloa Clúster de Cambio Global y Nuevas Energías (y, más concretamente, el sistema EOLO como recurso de computación de alto rendimiento HPC - High Performance Computing), infraestructura financiada por el Ministerio de Educación, Cultura y Deporte (MECD) y por el Ministerio de Economía y Competitividad (MINECO).

Referencias

[1] B. Krishnamurthy. “Privacy and Online Social Networks: can color less green ideas sleep furiosly?” IEEE Security and Privacy, Vol. 11, No. 3, pp. 14–20, May 2013.
[2] A. Westin. “Privacy and Freedom”, Vol. 25, New York: Atheneum: Washington and Lee Law Review, 1968.
[3] R. Gellman y P. Dixon. “Online Privacy: A Reference Handbook”, Santa Barbara, CA.: ABC - CLIO, September, 2011.
[4] R. Gross and A. Acquisti. “Information revelation and privacy in online social networks”, Proceedings of the 2005 ACM workshop on Privacy in the electronic society, Alexandria, VA, USA, pp. 71–80, November 2005.
[5] European Commission. “Press release: Privacy Enhancing Technologies (PETs)”, May 2, 2007.
[6] L. Fritsch. “State or the art of privacy-enhancing technology (PET)”, Norwegian Computing Center Report, Oslo, Norway, 2007.
[7] The META Group, “State of the art of privacy-enhancing technology (PET)”, Danish Ministry of Science, Technology and Innovation, Denmark, March, 2005.
[8] C. Adams. “A Classification for Privacy Techniques”, University of Ottawa Law and Technology Journal, Vol. 3, No. 1, pp. 35–52, 2006.
[9] A. Pfitzmann y M. Hansen. “Anonymity, unlinkability, unobservability, pseudonymity, and identity management: a consolidated proposal for terminology”, TU Dresden, February 2008.
[10] D. Solove. “A Taxonomy of Privacy”, University of Pennsylvania Law Review, Vol. 154, No. 3, January, 2006.
[11] C. Diaz y S. Gurses. “Understanding the landscape of privacy technologies”, Proc. of the Information Security Summit, pp. 58–63, Prague, Czech Republic, May, 2012.
[12] D. Chaum. “Untraceable electronic mail, return addresses, and digital pseudonyms”, Communications ACM, Vol. 24, No. 2, pp. 84–90, February 1981.
[13] M. K. Reiter y A. D. Rubin. “Crowds: anonymity for Web transactions”, ACM Transactions on Information Security and System Security (TISSEC), Vol. 1, No. 1, pp. 66-92, November 1998.
[14] B. Levine y C. Shields. “Hordes: a multicast based protocol for anonymity”, Journal of Computer Security, Vol. 10, No. 3, pp. 213-240, September 2002.
[15] C. Gulcu y G. Tsudik. “Mixing Email BABEL”, in Proceedings of the 1996 Symposium on Network and Distributed System Security, pp. 2-16, San Diego, CA, USA., February 1996.
[16] O. Berthold, H. Federrath y S. Kospel. “Web MIXes: A system for anonymous and unobservable Internet access”, in Proceedings of the International workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability, pp. 115-129, Berkeley, CA, USA., July 2000.
[17] D. Goldschlag, M. Reed y P. Syverson. “Hiding Routing Infomation”, in Proceedings of the the First Workshop on Information Hiding, pp. 137-150, London, UK, 1996.
[18] A. Back, I. Goldberg y A. Shostack. “Freedom systems 2.1. security issues and analysis”, Zero Knowledge Systems, May 2001.
[19] R. Dingledine, N. Mathewson y P. Syverson. “Tor: The second-generation onion router”, in Proceedings of the the 13th USENIX Security Syposium, pp. 303-320, San Diego, CA, USA, August 2004.
[20] K. Loesing. “Privacy-enhancing Technologies for Private Services”, University of Bamberg, 2009.
[21] M. Edman y B. Yener. “On Anonymity in an Electronic Society: A Survey of Anonymous Communication Systems”, ACM Computing Surveys, Vol. 42, No. 1, pp. 1–35, December 2009.
[22] G. Danezis. “Statistical disclosure attacks: Traffic confirmation in open environments”, in Proceedings of the Security and Privacy in the Age of Uncertainty Conference, (SEC2003), Kluwer, pp. 421–426, May 2003.
[23] J. F. Raymond. “Traffic Analysis: Protocols, Attacks, Design Issues, and Open Problems”, in Proceedings of the International Workshop on Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability”, New York, NY, USA, 2001.
[24] J. Portela García-Miguel, D. Rupérez Cañas, A. L. Sandoval Orozco, A. G. Silva Trujillo y L. J. García Villalba. “Ataque de Revelación de Identidades en un Sistema de Correo Electrónico”, Actas de la XII Reunión Española sobre Criptología y Seguridad de la Información (RECSI 2012), Donostia-San Sebastián, España, Septiembre 2012.

Herramienta para la Compensación de Parámetros de QoS y Seguridad

Ana Nieto, Javier Lopez
Departamento de Lenguajes y Ciencias de la Computación
Universidad de Málaga, España
Email:{nieto,jlm}@lcc.uma.es

***Resumen*—El análisis conjunto de mecanismos de seguridad y QoS es esencial para las redes heterogéneas donde diversos dispositivos pueden coexistir en entornos dinámicos. En concreto, los dispositivos no siempre pueden ser conocidos, por lo que diferentes requisitos y mecanismos pueden surgir para el análisis. En este artículo, proponemos una herramienta para facilitar la configuración de entornos basada en el análisis paramétrico de dependencias, tomando como base de conocimiento un conjunto de parámetros de seguridad y QoS. Esta forma de análisis de parámetros a alto nivel permite considerar las dependencias y la compensación entre mecanismos con independencia del sistema de información subyacente. Posibilita por tanto evaluar el impacto que tales mecanismos, y otros definidos acorde al modelo, tienen sobre un sistema previo a su despliegue.**

***Palabras clave*—CPRM; QoS; PRM; Seguridad;**

I. Introducción y Fundamentos

Diversos modelos para el análisis conjunto de aspectos de seguridad y calidad de servicio (QoS) emergen como consecuencia directa de la amplia diversidad de dispositivos que componen las redes heterogéneas. En particular, los modelos genéricos para el análisis del balanceo o compensación de requisitos de seguridad y QoS son, desde el punto de vista práctico, los más relevantes para las redes heterogéneas de composición dinámica, en las que no se puede prever con gran exactitud los dispositivos que formarán la red.

Definimos un modelo genérico para el análisis de la compensación de seguridad y QoS como aquel que se abstrae de detalles específicos de una tecnología y que ofrece la posibilidad de integrar en el estudio cualquier tipo de tecnología y dispositivo a distinto nivel. De hecho, podemos encontrar algunos ejemplos de modelos genéricos en la literatura que se ajustan en mayor o menor medida a esta definición [1], [2], enfoques más específicos sobre seguridad o QoS [8], [4], [5], [3], y otros, que emplean técnicas paramétricas para mejorar la configuración de servicios [10]. Por ejemplo, en [1] se emplean técnicas de *model checking* para verificar las equivalencias entre especificaciones de seguridad y QoS, con el objetivo de controlar los flujos de información ilegítimos en el sistema. No obstante, obliga a definir un modelo de comunicación entre las aplicaciones del sistema, restringiendo por tanto su ámbito de uso. Alternativamente, en [2] se define un modelo basado en el contexto, que proporciona una función de utilidad para tener en consideración las preferencias del usuario. Sin embargo, no permite medir el impacto que unos parámetros del sistema tienen sobre otros, y el conjunto de contextos es limitado.

No obstante, el análisis conjunto de los mecanismos de seguridad y QoS debería basarse en el estudio de relaciones paramétricas, es decir, relaciones de dependencia entre los parámetros que definen la composición de los mecanismos de seguridad y los de QoS. Además, definir estas relaciones en base a un contexto es básico para expresar la relevancia de los parámetros, relaciones, operaciones y otros componentes y propiedades, que tienen cabida en el sistema de información.

I-A. Definición de un Modelo para el Análisis de Relaciones Paramétricas basado en el Contexto (CPRM)

En base al paradigma actual y futura convergencia de las redes, en [6] definimos un modelo para estudiar las relaciones paramétricas basado en el contexto, denominado CPRM por sus siglas en inglés (*Context-based Parametric Relationship Model*). Dicho modelo define la estructura de un sistema en base a un conjunto de parámetros y sus relaciones, un conjunto de operaciones que definen efectos sobre los parámetros dependientes, y una estructura de pesos que define la relevancia subjetiva y no subjetiva de los componentes del modelo.

Por ejemplo, un administrador puede considerar subjetivamente que la confianza es un parámetro clave para la subsistencia del sistema de información. En ese caso, el parámetro confianza tendría un peso mayor en el sistema que otros parámetros menos relevantes dado el caso. A su vez, los mecanismos que implementen el valor de confianza podrían heredar la relevancia o peso de su parámetro padre, en este caso, el parámetro confianza. Estos valores subjetivos estarían sujetos a la variabilidad del contexto, de forma que en un momento dado, ya sea por las medidas de seguridad adoptadas o por el entorno donde está el individuo, su relevancia puede variar. Por ejemplo, en un entorno familiar bien definido, el parámetro confianza y los mecanismos estrechamente dependientes podrían relajar su relevancia de no existir otras dependencias que se lo impidan. Esto es así, porque en el contexto *hogar* el individuo podría asumir que la confianza viene dada por su ubicación. Aunque no tiene porqué ser así.

Además, el modelo también contempla valores no subjetivos; destinados a definir el impacto o reacción en cadena que podría ocasionar una dependencia. Estos valores, se definen, en primer lugar, de forma aproximada en las dependencias del contexto general (GC, *General Context*), mientras que, una vez que los parámetros son instanciados, el peso es actualizado al contexto particular (PC, *Particular Context*).

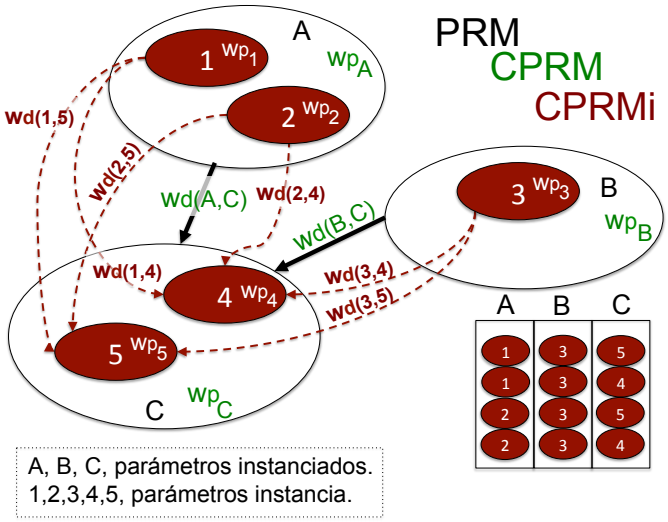


Figura 1. Instanciación de Parámetros.

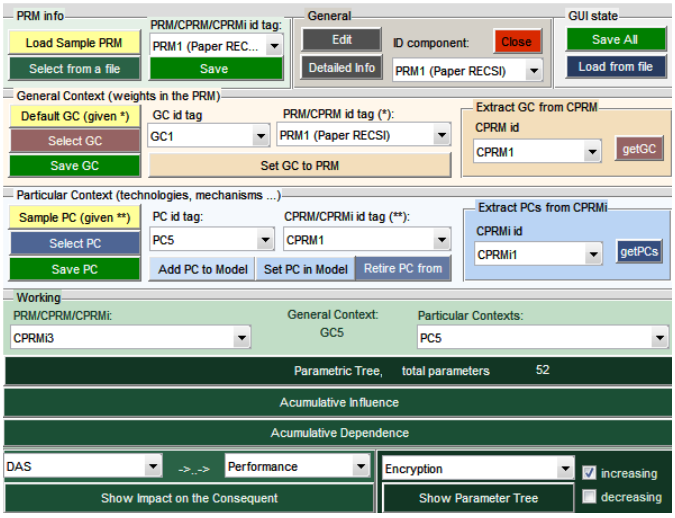


Figura 2. Interfaz de Administración.

La Figura 1 muestra parte de un sistema de dependencias paramétrico instanciado siguiendo el modelo CPRM dada su definición en [6][1]. La descripción de la formulación matemática asociada a las dependencias, así como las reglas de coherencia para la integración de contextos pueden consultarse en trabajos previos con más detalle [7]. En este caso, nos centramos en la visión general de cómo la integración de los parámetros y su instanciación quedarían reflejadas.

Partimos de un conjunto de parámetros que definen, de forma general, el escenario a evaluar. Este contexto base (BC) es fijo y no varía, y se encuentra en el PRM [2]. Pueden variar los pesos/relevancia de los parámetros, pero el BC siempre queda presente a la espera de que sus parámetros, relaciones, niveles, tipos y operaciones tomen los valores de contexto definidos en el GC y, posteriormente, en los sucesivos PCs. El BC es el resultado de un proceso de análisis exhaustivo sobre las arquitecturas y el entorno donde la herramienta tendrá cabida. En nuestro caso, surge del estudio de mecanismos de Seguridad y QoS en el *Internet del Futuro* [7]. Aunque la herramienta propuesta permite definir BC personalizados, siendo por tanto extensible a otros ámbitos de estudio, nuestro principal objetivo es su uso para el análisis de la compensación entre parámetros de Seguridad y QoS. En efecto, el BC que proporcionamos define dichos tipos de relaciones y no otros, que deberían ser agregados con posterioridad, según el caso.

En este artículo proporcionamos las directrices básicas para la implementación y el uso del modelo por medio de una herramienta desarrollada a tal efecto, a la que denominaremos SQT, por sus siglas en inglés *Security and QoS tradeoff Tool*. SQT proporciona un interfaz gráfico para la administración (Figura I-A) permitiendo al operador importar esquemas de modelo (PRM, CPRM, $CPRM_i$) y de contexto (GCs y PCs), salvar cualquier esquema en ficheros para su posterior uso y modificación, así como el espacio de trabajo completo, con los modelos y contextos asociados. También es posible extraer o eliminar contextos de los esquemas de modelo contextuales (CPRM, $CPRM_i$). El objetivo final es el análisis dirigido a la obtención de mediciones sobre el modelo de dependencias:

1. Incremento y decremento de parámetros.
2. Selección de conjuntos de parámetros por tipo y nivel.
3. Selección de parámetros instanciados (llamados padre) o bien de sus instancias (llamadas hijos) para distinguir entre diferentes opciones de configuración.
4. Calcular árboles de dependencias específicos para un parámetro, con el fin de posibilitar un examen más exhaustivo sobre el proceso de incremento/decremento.

A su vez, SQT permite visionar los resultados mediante diagramas de barras superpuestas que indican el impacto de un conjunto de parámetros en el resto de parámetros del sistema, o bien sobre un conjunto específico, en base al tipo/nivel, etc. También es posible seleccionar un parámetro en particular, como veremos en el caso de estudio. Otro modo de representación empleado es el uso de grafos, por medio de GraphViz. Así, el modelo de dependencias es representado mediante un grafo, en el que los parámetros se muestran acorde con la representación del tipo y agrupados por niveles según se define en el modelo.

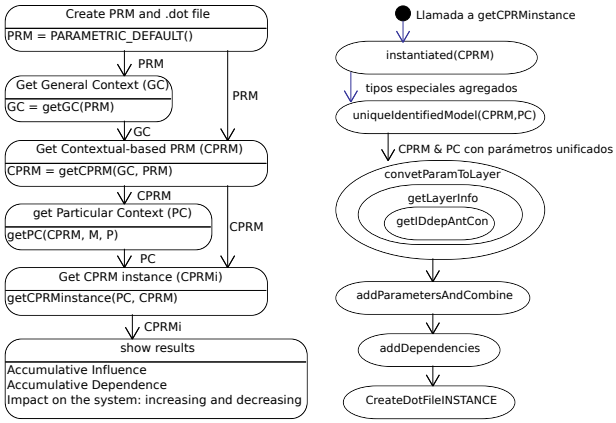
Para posibilitar el cumplimiento de tales requisitos, la herramienta implementa un conjunto de reglas de coherencia definidas para el modelo en [6]. El cumplimiento de estas reglas garantiza que el sistema paramétrico final mantiene la coherencia entre las dependencias.

El resto del artículo se divide como sigue. La Sección II muestra los detalles de implementación del prototipo conforme los requisitos dados. La Sección III estudia la usabilidad del prototipo para el análisis de la compensación entre parámetros de Seguridad y QoS. Por último, exponemos las conclusiones

[1]Hacemos referencia al modelo que define las estructuras PRM, CPRM, $CPRM_i$ (modelo instanciado a partir de un CPRM) y la relación entre sus componentes como CPRM.

[2]De cara a nuestro estudio, el BC no representa una estructura contextual, ya que los parámetros en el BC (en el PRM) carecen de pesos.

y el trabajo futuro.



(a) Generación $CPRMi$. (b) Integración de un PC.

Figura 3. Modelo de Componentes.

II. Prototipado del Modelo

El prototipo del modelo CPRM fue implementado en Matlab, ofreciendo una versión *plug-in*. Para el uso de SQT con toda su funcionalidad, debe instalarse GraphViz, a fin de interpretar los ficheros .dot que contienen las dependencias [3]. Los siguientes apartados abordan el diseño de SQT.

II-A. Modelo de Componentes

El diseño de SQT está basado en el uso de componentes, tanto desde el punto de vista arquitectural, como desde el punto de vista de la integración de contextos, considerados como componentes intercambiables. Así, teniendo en cuenta que en un CPRM puede existir un único GC, cualquier estructura de modelo puede ser ampliada/modificada usando SQT por medio de la agregación/sustitución de un GC, y de tantos PCs como sea preciso. Cuando en el CPRM se integra un PC, decimos que se genera una instancia del CPRM y lo denotamos como $CPRM_i$.

Así mismo, de cara a facilitar la tarea de análisis, es preciso que podamos volver a una versión anterior del modelo retirando el último contexto agregado, o construir nuevos contextos retirando alguno de los contextos integrados (no necesariamente el último). Esto es posible gracias a las reglas de integración y coherencia definidas para el modelo que implementa SQT [6], y a la definición de la cadena de integración y estructuras de datos descritas aquí.

Para permitir dicha funcionalidad, la integración de componentes se efectúa en SQT conforme al diagrama de actividad mostrado en la Figura 3(a), en el que se ilustra la creación de un $CPRM_i$ a partir de un PRM[4], para un caso de prueba. Para ello, creamos estructuras intermedias por defecto.

En particular, empleamos las funciones getGC y getPC para extraer o generar un contexto en base a una estructura paramétrica. En el caso de getGC, si la estructura introducida es un PRM, y, por consiguiente, sin contexto asociado, generará un GC asociado a los parámetros de la estructura. Si, por el contrario, recibe un CPRM o un $CPRM_i$, que son estructuras con un GC asociado, entonces devolverá el GC asociado a la estructura. De igual forma, getPC sólo devuelve los PC asociados a una estructura cuando están definidos, es decir, cuando la entrada es un $CPRM_i$. En otro caso, devolvería un PC aleatorio adecuado al tipo de estructura[5].

Por otra parte, las funciones getCPRM y getCPRMinstance asocian contextos con modelos. Es decir, getCPRM recibe un modelo y un GC que asignará al modelo. De esta asignación se obtiene un modelo paramétrico contextualizado (CPRM) coherente. El caso de getCPRMinstance es ligeramente distinto, ya que en un $CPRM_i$ varios PCs pueden coexistir. Aunque ambas funciones persiguen obtener una estructura nueva a partir de un modelo y un contexto, en el caso de getCPRMi se precisa un análisis mucho más exhaustivo.

Dado que un $CPRM_i$ es una instancia de un CPRM, se espera que sea una estructura dinámica, donde los PCs son intercambiados con mucha más frecuencia que un GC, que, aunque puede ser modificado, se asume que es una parte mucho más estable. Por tanto, el caso en el que diferentes parámetros se identifiquen igual en un $CPRM_i$ y un PC podría ser posible, dada la diversidad de escenarios que podrían definirse como PC. Estos casos deben considerarse para no solapar comportamientos de distintos parámetros. Este es sólo un ejemplo de los aspectos a contemplar en la integración de PCs en un $CPRM_i$.

La Figura 3(b) muestra de forma más detallada la secuencia de acciones realizadas por la función de integración de PCs, getCPRMinstance. Esta función recibe una estructura CPRM y la transforma en el primer paso para incluir los tipos y campos adicionales en una estructura $CPRM_i$. Si la estructura ya es un $CPRM_i$, entonces no realiza ningún cambio inicial, se considera que la estructura está instanciada.

El siguiente paso, es asegurarnos de que los identificadores de los parámetros en el modelo, ya un $CPRM_i$, no coinciden con los identificadores de los parámetros del PC. Tras este paso, obtenemos un modelo unificado, en el que los parámetros del modelo y el contexto significan lo mismo. Para estudiar la compensación paramétrica de los parámetros instanciados, getCPRMinstance convierte en niveles los parámetros padre, es decir, aquellos parámetros p para los que existen parámetros en PC que instancian a p (de forma matemática: $p|\exists p2 \in PC, p \in P(p2)$). Estos niveles contienen información de interés para, en caso de retirar el PC que provocó la instanciación, que el nivel asociado a un padre desaparezca y se restaure como parámetro sin instanciar.

Por último, se establecerán las dependencias que heredará el hijo, y se crearán aquellas necesarias para mantener el modelo

[3]Los ficheros .dot pueden ser modificados directamente o interpretados desde otras herramientas.

[4]En este ejemplo, el PRM es creado usando una función por defecto acorde a la definición del modelo.

[5]M y P indican el número de parámetros instancia que queremos que se generen por cada parámetro del modelo que se recibe como entrada.

coherente. Por ejemplo, si una instancia (hijo) se relaciona con un parámetro con el que el parámetro padre no tiene relación, se agregaría una nueva dependencia entre el parámetro padre y el parámetro con el que se relaciona el hijo (ej. Figura 1).

Finalmente, a nivel de análisis, todas las pruebas posibles sobre un PRM son posibles sobre estructuras CPRM o $CPRM_i$ (inclusive la interpretación mediante diagramas .dot). La diferencia sustancial, es que mientras que un PRM es estático, un CPRM presenta también una visión subjetiva del contexto de la red, dando más relevancia a unos parámetros, relaciones u operaciones conforme a las prioridades de administración o el conocimiento profundo de la red. Un $CPRM_i$, además, permite la integración de partes dinámicas en base a particularidades, contextos más variables y fugaces, pero también más específicos. Una vez que se conocen el conjunto de dispositivos tanto como para establecer sus dependencias y darles valores no subjetivos, sino próximos a la realidad, partes del GC pueden ser instanciadas con el PC.

II-B. Estructuras de Datos

Aunque en los ejemplos anteriores se mostró la creación de GCs y PCs por defecto en base a un modelo o estructura, cualquier estructura PRM, CPRM, $CPRM_i$, PC o GC tiene su formato predefinido con el que son creadas y empleadas.

La herramienta mantiene todas estas estructuras como parte de una estructura general, S, que gestiona los esquemas y contextos y que puede ser salvada, como espacio de trabajo.

Desde el punto de vista de la implementación, se puede considerar que S (Exp. 1) contiene el modelo de datos, compuesto por las estructuras de modelo y de contexto. En particular, la Tabla I muestra, grosso modo, las diferencias existentes entre los esquemas de modelo. Éstas, permiten identificar cuándo una estructura de modelo es un PRM, un CPRM ó un $CPRM_i$, y gestionar las operaciones definidas acorde al tipo de estructura y su definición. Las partes comunes entre los modelos, son las que posibilitan la integración basada en componentes. En particular, como parte del esquema PRM, las propiedades de niveles, tipos, operaciones y parámetros contienen elementos comunes como por ejemplo identificadores inequívocos, nombre (*string*), y forma de representación visual en los diagramas Matlab o GraphViz (color y forma). Además, cada parámetro, una vez calculadas sus dependencias con el resto, conserva la matriz de dependencias paramétrica, definida en [7], creada de forma recursiva, que define todas las relaciones de dependencia posibles que involucran al parámetro. Dichas matrices ocupan espacio en la estructura del PRM, a cambio de evitar el cálculo de mapas repetidas veces. Se obtienen a su vez de la matriz de dependencias general, donde se muestran todas las relaciones simples en su forma matricial binaria. Este conocimiento se extrae a su vez de las denominadas dependencias en bruto (DB), que expresan las relaciones $A \rightarrow B$ por medio del identificador del parámetro A, el de la operación de dependencia y el del parámetro B.

$$S = \{D1, D2, D3, D4, D5\}; \quad (1)$$
$$D1 = \#prm, nxtID, \{\{prm1, id, info, file\}, ...\}; \quad (2)$$
$$D2 = \#cprm, nxtID, \{\{cprm1, id, info, gcid, file\}, ...\}; \quad (3)$$
$$D3 = \#cprm_i, nxtID, \{\{cpmi1, id, info, gcid, pclist, file\}, ...\}; \quad (4)$$
$$pclist = [pcid1, pcid2, ...]; \quad (5)$$
$$D4 = \#gc, nxtID, \{\{gc1, gcid, info, file\}, ...\}; \quad (6)$$
$$D5 = \#pc, nxtID, \{\{pc1, gcid1, info, file\}, ...\}; \quad (7)$$

El esquema para el PRM sienta las bases de los esquemas definidos para el modelo CPRM y las instancias $CPRM_i$. No obstante, hay diferencias que, aunque sutiles en el esquema, suponen un cambio notorio en el proceso de cálculo de SQT. Así, mientras que la estructura CPRM supone un punto de inflexión entre un PRM y un modelo instanciado, los cambios realmente relevantes se producen de cara a la definición de un $CPRM_i$. Esto se debe en gran medida a dos factores clave: la definición de los tipos especiales para las instancias de parámetros y los parámetros instanciados, y la conversión puntual de parámetros como niveles. Estos factores, junto a la capacidad de restauración y modificación del modelo por medio de la eliminación y agregación de contextos, suponen un gran cambio respecto los modelos no instanciados, que quedan relegados a un desempeño más estático.

II-B1. Estructuras para los Contextos: A modo de ejemplo, mostramos a continuación dos esbozos de definiciones de GC (Exp. 8-14) y PC (Exp.15-17).

$$GC(1, 1:2) = \{NL\{id\_nivel1\ peso1;\ id\_nivel2\ peso2; ...\}\} \quad (8)$$
$$GC(2, 1:2) = \{NT\{id\_tipo1\ peso1;\ id\_tipo2\ peso2; ...\}\} \quad (9)$$
$$GC(3, 1:2) = \{NO\{id\_op1\ peso1;\ id\_op2\ peso2; ...\}\} \quad (10)$$
$$GC(4) = \{\}; \quad (11)$$
$$GC(5, 1:2) = \{NP, NProp\}; \quad (12)$$
$$GC(6:(5+NP), 1:NProp) = \{id\_param1\ peso1; ...\} \quad (13)$$
$$GC(6+NP, 1:2) = \{ND, \{id\_dep1\ peso1; ...\}\} \quad (14)$$

Las estructuras de contexto comparten algunos campos con las estructuras de modelo. Esto es preciso dado que las primeras pretenden efectuar cambios sobre los componentes de los modelos (parámetros, relaciones, tipos...). No obstante, los campos NProp y NP hacen referencia a la propia estructura de contexto, no a los campos del modelo. Es decir, las estructuras de contexto definen su propia forma de extensión. Por ejemplo, en la versión actual, la parte de definición de parámetros en un PC cuenta con 5 campos de propiedad (NProp=5): una lista de identificadores de parámetros padre (idPadres), el identificador del parámetro (que puede ser modificado si las reglas de integración lo demandan), el nombre del parámetro y el peso.

$$PC(1, 1:4) = \{NP, Nprop, ND, \{IDpc, descrip.\}\}; \quad (15)$$
$$PC(2:(1+NP), 1:Nprop) = \{idPadres, id, nombre, peso\}; \quad (16)$$
$$PC\{3+NP\} = \{idParamA, idOp, idParamB, peso; ...\}; \quad (17)$$

Dado un PC, cuando un CPRM es instanciado (Fig. 3(b)), se crean dos tipos especiales: *instance* e *instantiated*. Así, cuando

Tabla I
CAMPOS PARA LAS ESTRUCTURAS DE DATOS DE LOS ESQUEMAS DE MODELOS

Fila,Columna: Propósito	Definición PRM	CPRM (cambios sobre PRM)	$CPRM_i$ (cambios sobre CPRM)
1,1-2: Info. niveles	Número de niveles (NL) + Propiedades de Niveles	Agrega a las propiedades de cada nivel un peso w_l	Define niveles especiales para los parámetros instanciados
2,1-2: Info. tipos	Número de tipos (NT) + Propiedades de Tipos	Agrega a las propiedades de cada tipo un peso w_t	Agrega los dos tipos especiales: *instance* e *instantiated*
3,1-2: Info. operaciones	Número de operaciones (NO) + Propiedades de Operaciones	Agrega a las propiedades de cada operación un peso w_o	-
4,1-2: Otra información	Directorio por defecto (DD)	-	Agrega información isobre las instancias realizadas
5,1: NP	Propio del modelo		
5,2: NProp	5	6	6
5+NP,1-NProp: Parámetros	Propiedades de Parámetros	Agrega a las propiedades de cada parámetro un peso w_p	Los parámetros instanciados cambian su nivel por el nuevo creado como resultado de su instanciación
6+NP,1: Dependencias	Dependencias en bruto (DB) ó Matriz de dependencias procesada (MD)		
6+NP,2-3: Tras procesar DB	matriz de ceros NPxNP + DB	Matriz de costes NPxNP + DB	-

un parámetro sea instanciado y se cree un nivel a partir de éste, se etiquetará al parámetro como *instantiated* permitiendo aplicar las reglas de herencia para el cálculo del impacto paramédico. A su vez, cuando el parámetro es etiquetado como *instance*, se espera un identificador del PC que provocó la instanciación, y se tiene en cuenta que el parámetro es más dinámico que un parámetro que no sea instancia.

III. CASO DE USO Y EVALUACIÓN

En esta sección mostraremos cómo realizar pruebas para estimar la compensación entre requisitos de Seguridad y QoS.

III-A. Parámetros del Contexto Base

El ejemplo propuesto para el caso de análisis está basado en el funcionamiento de una red de sensores. Como tal, considera como parte del conjunto de parámetros del contexto base (BC) aquellos parámetros generales que pueden estar relacionados con una red de sensores, así como las relaciones entre éstos (consultar [7]). Adaptado al caso que nos ocupa, los parámetros del BC son mostrados en la Tabla II[6].

Aunque el GC por defecto para estos parámetros es inicialmente establecido con peso igual a 1 para todos los parámetros ($\forall p | p \in PRM, w_p = 1$), es posible establecer un GC subjetivo, basado en nuestras prioridades de administración. Por ejemplo, aumentar la relevancia/impacto del cifrado (*Encryption*), de tal forma, que todos los parámetros que tengan una dependencia en la que *Encryption* se encuentre en el antecedente serán más afectados que el resto de parámetros. Los parámetros afectados por el incremento del parámetro *Encryption*, pueden consultarse usando el árbol paramétrico particularizado para un parámetro. El efecto, sin embargo, podrá variar dependiendo del tipo de relación definida entre los parámetros y de los pesos definidos para las relaciones. Por ahora, todos los pesos para las relaciones tienen valor unitario ($w_d = 1, \forall d : A \rightarrow B | d \in PRM$). Estos pesos pueden modificarse con un GC, pero en nuestro caso lo haremos con un ejemplo de instanciación de parámetros.

[6]Las dependencias entre los parámetros no son mostradas debido a su extensión. Puede consultarse el diagrama ampliado que contiene estos y otros parámetros en [7].

Tabla II
PARÁMETROS DEL CONTEXTO BASE (BC)

	HIGH-LEVEL REQUIREMENTS
QoS	Reliability, Fault Tolerance, Availability
Security	Authentication, Authorization, Confidentiality, Integrity, Trust, Privacy
	LOCAL PROPERTIES
Resources	PowerConsumption, Memory, Rayleigh Channel, Energy, ComputationTime
Security	Anti-Tampering, Encryption, Public Key Cryptography, Symmetric Cryptography, Secure Key Exchange, Secure Key redistribution, Key Generation, Signature Scheme
	COMMUNICATION
QoS	Data Rate, Packet Size, Signal Strength, Data Transmission, Transmission Time, Transmission Power
Characteristics	Time-sleeping, Required-time-on
Consequence	Retransmmission
	MEASUREMENTS
QoS	Throughput, Delay, Jitter, Packet Loss, Response Time, Bit Error Rate (BER)
	ENVIRONMENT
QoS	Allowable Bandwidth, Error Probability
Attacks	DoS, Malicious Devices
Consequence	Interference, Congestion, Overhead, Fading, Shadowing, Noise

III-B. Agregación de un Contexto

Una vez aplicado el GC, podemos aplicar diferentes PCs sobre el CPRM resultante. Este hecho conduce a lo que denominamos *instanciación del modelo paramétrico*. A modo de ejemplo, mostraremos los cambios producidos en el sistema al aplicar el contexto particular mostrado en la Tabla III, cuyos pesos son estimaciones acorde al trabajo [9].

Tabla III
PESOS w_d CONFORME [9]

	Dependence			Weight
General Parameter	Antecedent	R	Consequent	w_d
Authentication	CAS	+	ECDSA	1
	DAS	+	ECDSA	1
	CAS	$\neg c$	Memory	0
	DAS	$\neg c$	Memory	5
	CAS	c	PacketSize	5
	DAS	c	PacketSize	1
Signature Scheme	ECDSA	$\neg c$	Energy	1
	PairingBased	$\neg c$	Energy	5
	ECDSA	c	Computation Time	1
	PairingBased	c	ComputationTime	5

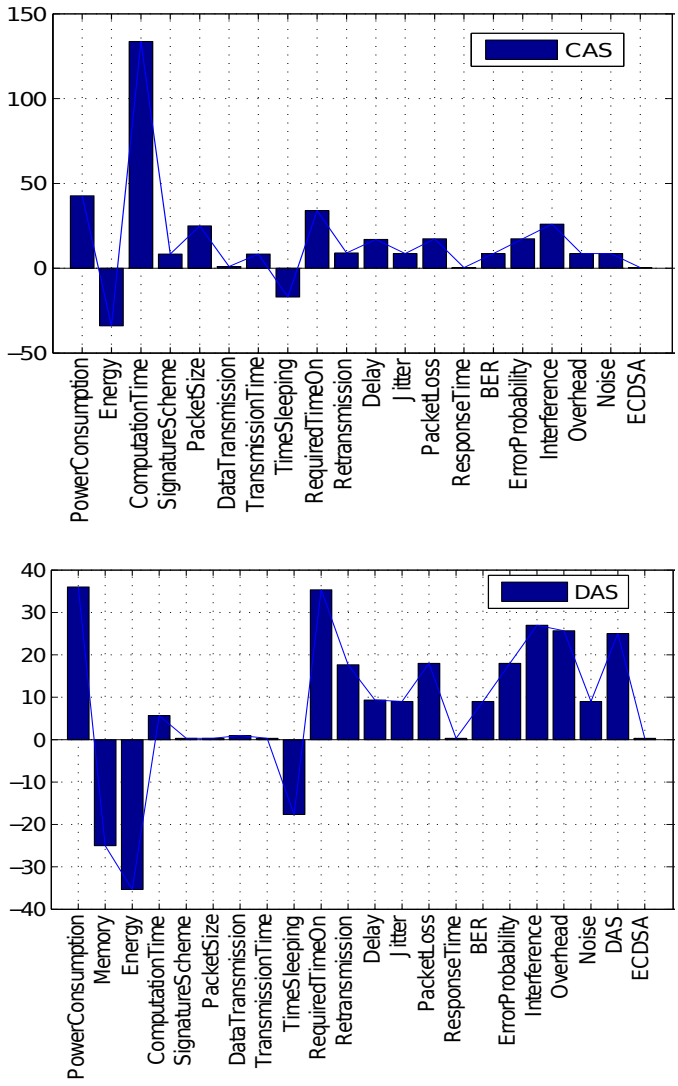


Figura 4. Impacto de CAS y DAS sobre el Rendimiento.

La Tabla III muestra los parámetros generales que serán instanciados (Authentication y SignatureScheme) y los parámetros instancia (CAS, DAS, ECDSA, PairingBased). Una vez integrado el nuevo contexto, los parámetros de Authentication y SignatureSheme pasarían a ser niveles, y como tales pueden ser consultados. Esta es una ventaja adicional del modelo, ya que permite comprobar el efecto que este último cambio de contexto tuvo sobre parámetros que ya se encontraban en el modelo. En el nuevo contexto final, cada vez que se incremente el parámetro Authentication o SignatureScheme, también serán incrementados los parámetros instancia, y con ellos los parámetros dependientes de éstos, que han podido introducir nuevas dependencias para hacer el modelo coherente.

Finalmente, el proceso de ajuste entre parámetros de Seguridad y QoS, se realiza en base al BC definido y la instanciación del modelo con los mecanismos cuyo impacto en el sistema resultante queremos medir. Por ejemplo, una vez introducido el último contexto (Tabla III), podemos evaluar el impacto que los mecanismos de autenticación CAS y DAS tienen sobre los parámetros de rendimiento (Figura 4) o de cualquier otro tipo. Note que los parámetros sobre los que se percibe el efecto no fueron obtenidos de [9], sino que son resultado de la integración con el BC definido a priori. La información del sistema será mucho más fiable y enriquecedora conforme el número de PC integrados sea mayor.

IV. Conclusiones y trabajo futuro

En este artículo proporcionamos las directrices básicas para la implementación y el uso de una herramienta para la evalucación de la compensación entre parámetros de Seguridad y QoS (SQT). SQT está basada en un modelo genérico para la compensación paramétrica basado en el contexto (CPRM) definido en trabajos previos. Un requisito importante perseguido es que cualquier contexto pueda ser intercambiado en un CPRM por otro nuevo o modificado. El caso de estudio abordado muestra cómo es posible emplear SQT para los fines propuestos.

Agradecimientos

Este trabajo ha sido parcialmente subvencionado por el Ministerio de Economía y Competitividad a través del proyecto ARES (CSD2007-00004). Adicionalmente, ha sido financiado por la Junta de Andalucía través del proyecto FISICCO (TIC-07223). El primer autor ha sido subvencionado por el Programa FPI.

Referencias

[1] Alessandro Aldini and Marco Bernardo. A formal approach to the integrated analysis of security and qos. *Reliability Engineering & System Safety*, 92(11):1503–1520, 2007.

[2] Mourad Alia, Marc Lacoste, Ruan He, and Frank Eliassen. Putting together qos and security in autonomic pervasive systems. In *Proceedings of the 6th ACM workshop on QoS and security for wireless and mobile networks*, pages 19–28. ACM, 2010.

[3] Siegfried Benkner and Gerhard Engelbrecht. A generic qos infrastructure for grid web services. In *Telecommunications, 2006. AICT-ICIW'06. International Conference on Internet and Web Applications and Services/Advanced International Conference on*, pages 141–141. IEEE, 2006.

[4] Roland Bless and M Rohricht. Secure signaling in next generation networks with nsis. In *Communications, 2009. ICC'09. IEEE International Conference on*, pages 1–6. IEEE, 2009.

[5] Cynthia Irvine and Timothy Levin. Toward a taxonomy and costing method for security services. In *Computer Security Applications Conference, 1999.(ACSAC'99) Proceedings. 15th Annual*, pages 183–188. IEEE, 1999.

[6] Ana Nieto and Javier Lopez. A context-based parametric relationship model (cprm) to measure the security and qos tradeoff in configurable environment. In *IEEE International Conference on Communications (ICC)*, pages 755–760. IEEE, 2014.

[7] Ana Nieto and Javier Lopez. Analysis and taxonomy of security/qos tradeoff solutions for the future internet. *Security and Communication Networks*, In Press.

[8] Tarik Taleb, Yassine Hadjadj Aoul, and Abderrahim Benslimane. Integrating security with qos in next generation networks. In *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pages 1–5. IEEE, 2010.

[9] Rehana Yasmin, Eike Ritter, and Guilin Wang. An authentication framework for wireless sensor networks using identity-based signatures. In *Computer and Information Technology (CIT), 2010 IEEE 10th International Conference on*, pages 882–889. IEEE, 2010.

[10] I-Ling Yen, Hui Ma, Farokh B Bastani, Hong Mei, et al. Qos-reconfigurable web services and compositions for high-assurance systems. 2008.

Monitorización y selección de incidentes en seguridad de redes mediante EDA

J. Camacho, G. Maciá-Fernández, J. Díaz-Verdejo, P. García-Teodoro
Departamento de Teoría de la Señal, Telemática y Comunicaciones - CITIC
Universidad de Granada
Email: {josecamacho, jedv, gmacia, pgteodor}@ugr.es

***Resumen*—Uno de los mayores retos a los que se enfrentan los sistemas de monitorización de seguridad en redes es el gran volumen de datos de diversa naturaleza y relevancia que deben procesar para su presentación adecuada al equipo administrador del sistema, tratando de incorporar la información semántica más relevante. En este artículo se propone la aplicación de herramientas derivadas de técnicas de análisis exploratorio de datos para la selección de los eventos críticos en los que el administrador debe focalizar su atención. Adicionalmente, estas herramientas son capaces de proporcionar información semántica en relación a los elementos involucrados y su grado de implicación en los eventos seleccionados. La propuesta se presenta y evalúa utilizando el desafío VAST 2012 como caso de estudio, obteniéndose resultados altamente satisfactorios.**

***Palabras clave*—análisis exploratorio de datos (*exploratory data analysis*), *big data*, visualización de datos (*data visualization*), seguridad en redes (*network security*), sistemas de monitorización de seguridad en redes (*network security monitoring systems*)**

I. Introducción

Los sistemas de monitorización de la seguridad en redes (NSM, del inglés *Network Security Monitoring*) [1] tienen como finalidad la agregación y análisis de los datos procedentes de los diversos mecanismos y sensores desplegados en el entorno de red, a fin de validar y, en su caso, responder a incidentes de seguridad. Aunque suelen incorporar datos procedentes de sistemas de detección de intrusiones (IDS, del inglés *Intrusion Detection Systems*) [2] como elemento relevante, no son, en sí mismos, sistemas IDS. Por el contrario, su operación está orientada a seleccionar, priorizar y validar las alertas generadas por otros sistemas de monitorización y trazado de eventos.

Entre las limitaciones que deben afrontar los NSM podemos mencionar el gran volumen de datos que deben manejar, ya que integran información de múltiples fuentes, muchas de ellas generando un elevado número de registros (p.e., trazas de cortafuegos, de sesiones, alertas de IDS, etc.). Adicionalmente, los datos deben ser preprocesados, agregados y presentados al administrador de forma que éste pueda comprenderlos y gestionarlos fácilmente. En consecuencia, dos son los retos más relevantes para el diseño de NSM: el análisis de los datos y la presentación/visualización de los resultados.

La mayoría de los NMS existentes (p.e., Sguil[1] o Snorby[2]) se limitan básicamente a recopilar e interrelacionar los datos procedentes de los sensores con la finalidad de facilitar su análisis y consulta por parte del administrador, mostrándolos en base a secuencias temporales y/o priorizándolos a partir de esquemas simples. En algunos casos se incluyen algunas heurísticas y estadísticas simples (p.e., Pravail Security Analytics[3]), pero es evidente que se requieren métodos y técnicas más potentes y de mayores prestaciones para el análisis y visualización de los datos. En este contexto, las técnicas de análisis exploratorio de datos (EDA) [3] pueden resultar extremadamente útiles tanto para establecer los eventos relevantes en los que el administrador debería centrar su atención, como para mostrar las propiedades o características implicadas en cada evento.

En este trabajo proponemos y evaluamos una metodología basada en EDA que proporciona medidas y gráficas para conseguir el objetivo antes mencionado. Para ello se realiza una elección de herramientas que, secuenciadas adecuadamente, permiten, en primer lugar, determinar los eventos potencialmente relevantes sin intervención del administrador. A partir de estos, mediante la obtención e interpretación de algunas gráficas, el administrador puede recabar información semántica respecto de dichos eventos que puede serle de utilidad para la posterior comprobación o supervisión de los mismos.

El resto del artículo se estructura como sigue. En la Sección II se presentan brevemente las herramientas y técnicas en las que se basan los análisis de datos subsiguientes. En la Sección III se describe el funcionamiento del sistema propuesto, explicitándose la secuenciación de las técnicas y procedimientos a aplicar. En la Sección IV se describe la aplicación del sistema desarrollado al reto VAST 2012 [4], para lo que se describirá previamente dicho reto así como la parametrización realizada, aspecto clave del análisis. Finalmente, en la Sección V se presentan las contribuciones más relevantes del trabajo y se apuntan brevemente algunos trabajos de futuro.

II. Herramientas de análisis exploratorio de datos

El análisis exploratorio de datos (EDA) tiene como objetivo facilitar el conocimiento y visualización de la estructura que

[1] http://sguil.sourceforge.net
[2] https://snorby.org
[3] http://www.arbornetworks.com/products/pravail/securityanalytics

presenta un conjunto de datos. Para ello utiliza una serie de técnicas y herramientas que permiten analizar su propiedades relevantes y presentarlas de forma adecuada para facilitar su interpretación. Entre las técnicas empleadas se encuentran algunas bien conocidas como PCA (*Principal Component Analysis*) [6], así como otras más novedosas, propuestas recientemente por parte de los autores, como son MEDA [7] y oMEDA [8].

A continuación, se describen brevemente las técnicas utilizadas:

- PCA: El análisis de componentes principales permite transformar un conjunto de N observaciones, cada una de ellas con M variables o componentes que pueden estar correlacionadas entre sí, a un nuevo espacio de características decorrelacionadas denominadas componentes principales (*Principal Components* o *PCs*). Sin entrar en detalles, que pueden consultarse en [5], si $\mathbf{X}$ es la matriz de datos, de dimensión $N \times M$, el análisis PCA permite expresar estas observaciones de acuerdo a:

$$\mathbf{X} = \mathbf{T}_A \cdot \mathbf{P}_A^t + \mathbf{E}_A, \quad (1)$$

  donde A es el número de PCs incluidas en el modelo, $\mathbf{T}_A$ es la matriz $N \times A$ de puntuaciones (*scores*), $\mathbf{P}_A$ la matriz $M \times A$ de cargas (*loadings*), compuesta por los A autovectores de $\mathbf{XX} := \mathbf{X}' \cdot \mathbf{X}$ con los mayores autovalores asociados, y $\mathbf{E}_A$ la matriz $N \times M$ de residuos. En el contexto del análisis de datos podemos decir, de forma coloquial, que el objetivo de este análisis es retener la mayor información posible sobre los datos con el menor número posible de parámetros. El procedimiento para la selección adecuada de A depende de la aplicación concreta considerada [9]. El resto de herramientas se basan en el modelo PCA.
- Gráficos de evolución: Los gráficos de evolución, utilizados ampliamente en el entorno industrial, permiten visualizar de forma simple la parte del modelo y de los residuos obtenida en la ec. (1) para el conjunto de observaciones. Para ello, se obtienen una pareja de gráficos a partir del *leverage* o estadístico T^2 de Hotelling, que comprime la información en el modelo, y la estadística Q [10], que comprime la información en el residuo. En el contexto de la seguridad, ambas gráficas permiten identificar con sencillez cualquier evento anómalo.
- MEDA: Los gráficos MEDA son mapas de color de tamaño $M \times M$ en los que se representa la relación (positiva o negativa) existente entre las parejas de variables de un conjunto de datos. Los coeficientes de MEDA son una variante de la correlación menos sensible al ruido y, por tanto, con mejores cualidades para detectar la estructura en los datos. Para facilitar la visualización de los gráficos MEDA, se suele usar un método de serialización [11] que reordena las variables de acuerdo a un criterio de similitud. De esta forma es más fácil identificar grupos de variables, ya que tienden a formar cuadrados en el gráfico. En el contexto de la seguridad, los grupos de variables nos permiten identificar los tipos de tráfico o incidentes que tienen lugar en la red.
- oMEDA: Los gráficos oMEDA permiten comparar valores de variables en dos grupos de observaciones a partir de un diagrama de barras. Así, un valor positivo para una variable en oMEDA significa que el primer grupo de observaciones presenta un valor mayor para dicha variable que el segundo grupo, mientras que un valor negativo representa lo contrario. En el contexto de la seguridad, oMEDA se utiliza para identificar las variables relacionadas con un evento anómalo, comparando dicho evento con la tendencia genérica en la red. El resultado nos permite determinar características del evento anómalo, que potencialmente nos pueden permitir identificar las causas de dicho evento y, en su caso, proponer medidas paliativas o de respuesta de forma veloz y eficaz.

Las herramientas descritas se encuentran implementadas en un módulo para ©Matlab desarrollado por uno de los autores [12].

III. Monitorización y visualización: arquitectura y metodología

La metodología de monitorización y visualización de incidentes de seguridad propuesta se basa en la detección e interpretación de anomalías a partir del análisis PCA, para lo que se usan las gráficas de *Hotelling* T^2 *y* Q, junto con MEDA y oMEDA para determinar las variables y relaciones entre ellas asociadas a dichas anomalías. Esta combinación de herramientas resulta extremadamente útil para los fines mencionados en escenarios caracterizados por un elevado número de datos y parámetros.

En la Figura 1 se muestra un diagrama de bloques del sistema planteado para NSM. Como puede observarse, se consideran dos bloques diferenciados que se discuten a continuación.

III-A. Preprocesado

En este bloque se preparan los datos procedentes de las fuentes para su análisis. Las secuencias de datos de entrada son preprocesadas y parametrizadas de acuerdo a un conjunto de características/variables seleccionadas.

Cada variable contabiliza el número de veces que, durante un cierto intervalo de tiempo w, aparece cierto valor o valores en los registros (*logs*) del dispositivo fuente. A modo de ejemplo, una variable podría contabilizar el número de veces que un puerto determinado, p.e., el 21 (ftp), aparece en las trazas durante un periodo de 1 minuto. La motivacion para esta elección es que el número de entradas en una traza en las que aparece un puerto concreto puede proporcionar información para detectar eventos asociados a un protocolo.

Aunque el sistema de parametrización puede diseñarse de forma específica para los dispositivos de monitorización y detección disponibles en la red, se pueden definir ciertas buenas prácticas de diseño:

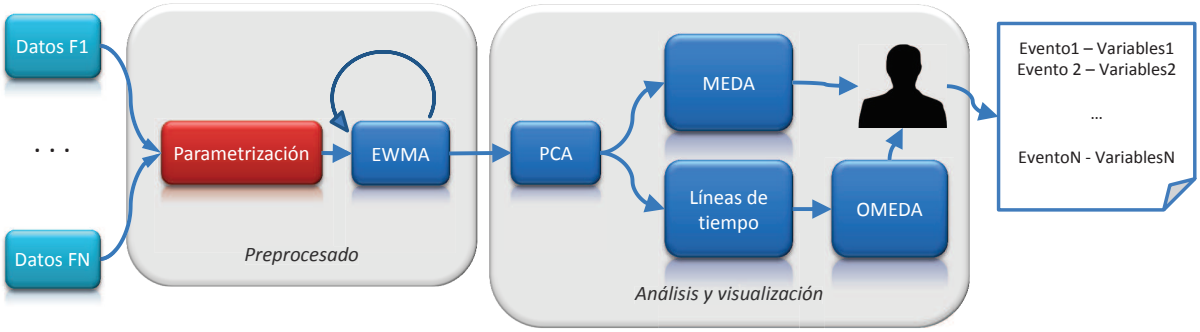


Figura 1. Diagrama de bloques del sistema.

- Seleccionar una variable por nivel de seguridad, prioridad o alarma en los registros del dispositivo (por ejemplo, variable "*warning*z variable "*critical*").
- Seleccionar una variable por código en los registros del dispositivo (por ejemplo, variable contador del código ASA-4).
- Seleccionar una variable por dirección IP o grupo de direcciones IP con sentido topológico o funcional en la red (por ejemplo, variable contador de la IP del DNS local o de las IPs de un departamento concreto).
- Seleccionar una variable por grupo de puertos relacionados con un protocolo de interés (por ejemplo, variable contador del puerto 80 y el 8080).

En cada intervalo de tiempo w, se combinan las características evaluadas para cada fuente en un único vector, $\overrightarrow{x_t}$, que será la observación correspondiente al instante t sobre la que se realizarán los análisis posteriores. A continuación, esta observación se utiliza para actualizar, siguiendo una estrategia de media móvil de peso exponencial o EWMA (*Exponentially Weighted Moving Average*), la matriz $\mathbf{XX}$ que representa el estado actual de la red monitorizada. La matriz se actualiza con la entrada de nuevos datos de la forma $\mathbf{XX}_t = \lambda \cdot \mathbf{XX}_t + \overrightarrow{x_t}' \cdot \overrightarrow{x_t}$, donde $0 \leq \lambda \leq 1$ es un factor de olvido que permite descartar información pasada.

III-B. Análisis y visualización

En el segundo bloque se realizan todas las operaciones necesarias para el análisis de los datos, visualización y posterior interpretación. Se inicia el procesamiento realizando un análisis PCA de $\mathbf{XX}$. La aproximación utilizada para la parametrización, a diferencia de las habitualmente utilizadas en los NSM, puede generar un elevado número de parámetros y, consecuentemente, una alta dimensionalidad de las observaciones. Sin embargo, esto no supone un problema dado el análisis PCA que se realiza a continuación. Usando PCA, el sistema permite identificar eventos donde se correlacionan las variables contador antes mencionadas, permitiendo de forma sencilla establecer puertos, segmentos de red, niveles de seguridad vulnerados en firewall o IDS, etc., asociados a cada evento anómalo.

A partir del modelo PCA se obtiene la estructura de las variables con los gráficos MEDA y la evolución temporal (líneas de tiempo) de los estadísticos *Hotelling* T^2 y Q, que serán utilizados para detectar anomalías, las cuales se reflejan en estos gráficos por picos en la evolución. Para cada anomalía o conjunto de anomalías próximas en el tiempo se obtienen gráficos oMEDA para determinar cuáles son las variables relacionadas con dicha anomalía.

De la metodología propuesta resulta relevante su capacidad, no sólo para manejar grandes volúmenes de datos, sino también para gestionar una alta dimensionalidad. Es decir, los eventos u observaciones del sistema pueden ser representados con tantos parámetros como se estime oportuno, no siendo problemática la introducción de información redundante o relacionada, que será adecuadamente procesada por los esquemas PCA subyacentes. Por el contrario, cuantos más parámetros se incluyan, mayor sera la información que podrá extraerse. Esta es una característica diferencial de la propuesta, ya que la mayoría de las herramientas de análisis de redes operan sobre series de datos unidimensionales o de reducida dimensionalidad [13].

IV. Caso de estudio: aplicación a VAST 2012

La mejor forma de explicitar y mostrar las potencialidades de la metodología propuesta en la sección anterior es aplicarla y explicarla en un escenario concreto. Para ello consideraremos el segundo reto del VAST 2012 [4].

Este reto considera un escenario correspondiente a una red corporativa bancaria con varias sedes y acceso a Internet (Figura 2) en la que ocurren incidentes de seguridad durante dos días. El desafío consiste en determinar los eventos más relevantes, sus causas y las posibles soluciones.

Los datos proporcionados consisten en una traza de un cortafuegos Cisco ASA, conteniendo 23.711.341 registros, y la salida generada por un IDS, que incluye 35.948 registros. Los conjuntos de datos, su descripción y los detalles sobre el reto se encuentran disponibles en [4].

IV-A. Parametrización y preprocesado

De acuerdo a la metodología propuesta, los datos procedentes de las trazas del IDS y del cortafuegos se han

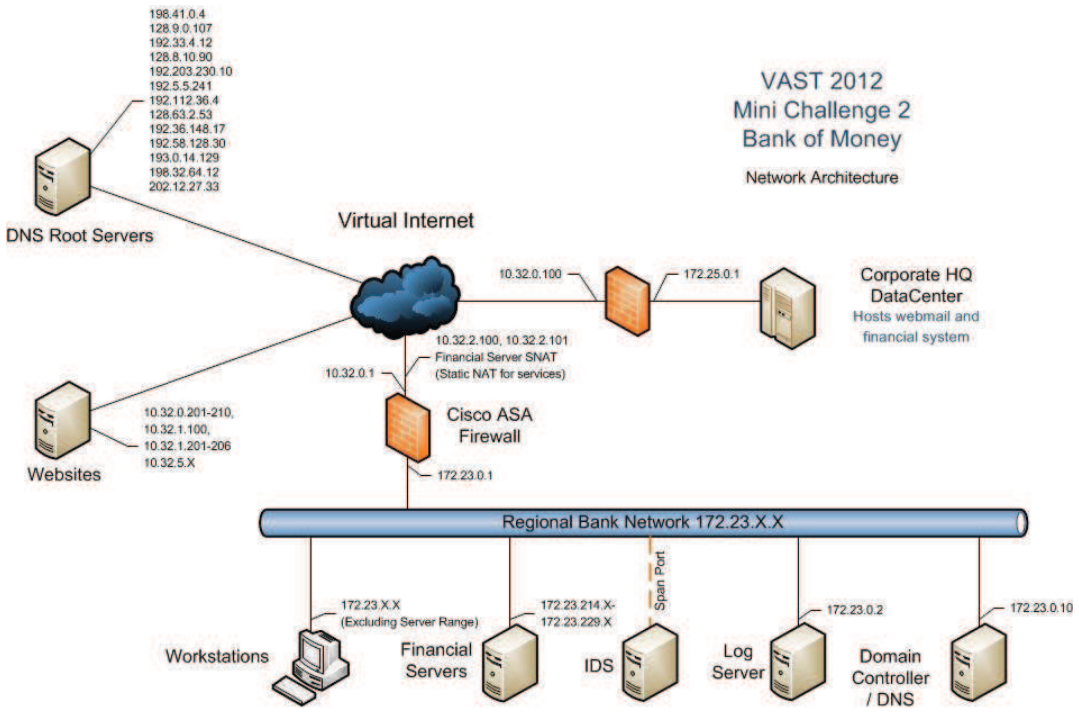


Figura 2. Red utilizada en el reto VAST 2012.

parametrizado agregando los datos durante periodos de $w = 1$ minuto. La elección de la ventana temporal viene determinada por la resolución temporal de las trazas del IDS, que impiden el uso de ventanas de menor tamaño. Se obtienen así 2.350 observaciones, ordenadas temporalmente.

Cada observación del sistema corresponde a un vector de 112 características o variables que representa la información procedente de ambas fuentes de datos. En particular, se han asignado 69 variables para las trazas del cortafuegos y las restantes 43 para las trazas del IDS. En la Tabla I se muestran los números de parámetros establecidos para cada campo presente en las trazas. Así, se consideran 17 parámetros asociados a cada uno de los puertos correspondientes a servicios estándar (número de puerto inferior a 1024) que aparecen en las trazas. En el caso de las direcciones IP, se han establecido 9 parámetros a partir de la topología de la red y de los rangos de direcciones existentes. En la Tabla II se muestran algunos de los parámetros seleccionados. La notación utilizada hace referencia a la fuente de los datos (*fw* o *ids*) y al significado o *flag* asociado a cada uno.

Tabla I
NÚMERO DE PARÁMETROS DEFINIDOS PARA CADA TIPO DE CARACTERÍSTICA.

	Campo	#parámetros
Trazas cortafuegos	Prioridad syslog	5
	Operación	6
	Código del mensaje	25
	Protocolo	3
	Dirección IP	9
	Puerto	17
	Dirección	2
	Tiempos conexión	2
	Subtotal	**69**
Trazas IDS	Dirección IP	9
	Puerto	17
	Tipo alerta	5
	Prioridad	3
	Etiqueta	9
	Subtotal	**43**

IV-B. Análisis y visualización

Una vez realizada la parametrización de los datos de entrada se procede a realizar un análisis PCA que será la base para el resto del estudio. A partir del modelo PCA, se obtiene un gráfico MEDA (Figura 3) que muestra la existencia de agrupaciones de variables en el conjunto de datos (cuadrados rojos). MEDA nos permite establecer, a nivel general, las relaciones comunes entre variables en nuestra red. A modo de ejemplo, uno de los cuadrados rojos relaciona *logs* de prioridad media en el IDS (`ids_prio2`) reportando intentos de robo de información (`ids_leak`) en el firewall (`ids_ipfwhq`) utilizando el protocolo VNC (`ids_lvnc`).

MEDA nos da una idea de eventos de seguridad comunes en nuestra red, pero no incorpora información temporal. La evolución temporal se analiza con los gráficos de evolución, Figura 4, donde las posibles anomalías se identifican como los

Tabla II
EJEMPLOS DE PARÁMETROS UTILIZADOS Y VALORES ASOCIADOS.

Parámetro	Campo	Valor(es) asociado(s)
fw_syscritical	Prioridad syslog	Critical
fw_syserror	Prioridad syslog	Error
fw_as37	Código mensaje	asa-3-710003
fw_pshell	Puerto	514
ids_ipfwr	Dirección IP	10.32.0.100 o 172.25.0.1
ids_iplog	Dirección IP	172.23.0.2
ids_misc	Clasificación	Misc. activity

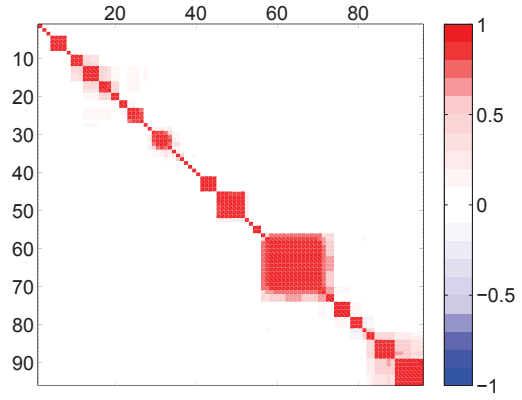


Figura 3. MEDA para todo el conjunto de datos.

valores más altos.

Las observaciones seleccionadas como anomalías son analizadas, bien en solitario, bien en grupos de observaciones consecutivas, para determinar las variables que hacen de ellas valores anómalos. A este fin, como se indicó en la Sección III, se obtienen gráficos oMEDA.

Para ilustrar el procedimiento, consideraremos el gráfico oMEDA de las observaciones {1,11,13,15} (Figura 5). A partir de esta gráfica se identifican dos parámetros con valores muy altos: *fw_iplog* (variable 54) y fw_syslog (variable 55). De acuerdo a esta información, el administrador puede concluir que las anomalías se encuentran relacionadas con el puerto *syslog* en el servidor de trazas *fw_iplog*. En el caso de las observaciones {374, 375}, dos de las que mayores valores proporcionan en las líneas de tiempo, se identifican de forma análoga las variables *ids_lssh, ids_pssh, fw_ptelnet, ids_limap, ids_lpop3, ids_leak, ids_ipfwhq* e *ids_prio2* como asociadas a la anomalía. Esto apunta a la existencia de problemas relacionados con intentos de acceso o fuga de información en los servicios SSH, IMAP y POP. El análisis manual de las trazas para el periodo de tiempo asociado a las observaciones nos lleva a la conclusión de que en este periodo se producen escaneos e intentos reiterados de acceso en los puertos correspondientes, lo que resulta coherente con la información proporcionada por el sistema.

IV-C. Resultados

A partir de la información obtenida en los pasos previos, tanto a nivel de observaciones a supervisar como de las variables implicadas en cada caso, se ha procedido a la

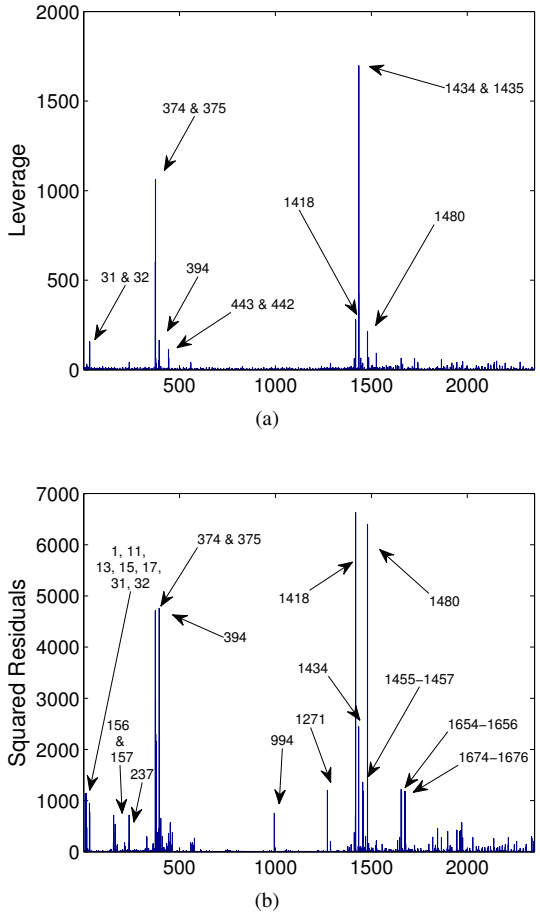


Figura 4. Evolución temporal de *leverage* (a) y residuo (b).

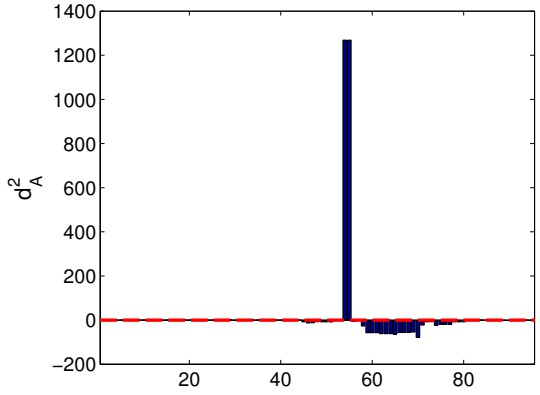


Figura 5. oMEDA para las observaciones {1,11,13,15}

inspección e intrepretación de los registros asociados a dichas observaciones, tanto en las trazas de IDS como de cortafuegos. Los detalles de dicho análisis exceden los objetivos del presente artículo, por lo que a continuación nos limitaremos a relacionar los incidentes encontrados y a comparar nuestros hallazgos con los de otros autores participantes en el reto, incluyendo los ganadores [14] (Tabla III).

Como resultado del análisis realizado, se han identificado

Tabla III
INCIDENTES DE SEGURIDAD REPORTADOS POR NUESTRO SISTEMA Y POR LOS DE OTROS AUTORES.

Anomalía	Propuesta	[14]	[15]	[16]
Ataques DNS/Controlador	X			
Intentos de intrusión al cortafuegos	X	X	X	
Tráfico FTP hacia nodos externos	X			X
Actividad IRC	X	X	X	
Errores en trazas	X			

las siguientes actividades sospechosas, para las que también se han obtenido los intervalos de actividad:

- Ejecución remota no interactiva.
- *DNS spoofing* hacia el servidor DNS y el controlador de dominio. Esta actividad se circunscribe a la red interna.
- Escaneo de puertos en el cortafuegos.
- Ataques de *buffer overflow* y denegación de servicio hacia el servidor DNS y el controlador de dominio.
- Actividad IRC continuada.

Adicionalmente, algunas de las anomalías encontradas han resultado en la constatación de errores de formato en los archivos de traza en un volumen no despreciable. Estos errores no habían sido informados por ninguno de los autores participantes en el reto ni en publicaciones posteriores.

Se puede comprobar (Tabla III) que el sistema propuesto ha permitido identificar no sólo los eventos previamente hallados por otros autores, sino algunos nuevos (ataques DNS y errores en las trazas) a partir del análisis de un reducido número de observaciones seleccionadas por el mismo. Adicionalmente, dicho análisis ha sido realizado de forma dirigida, focalizando la atención en las variables sugeridas a partir de la metodología propuesta.

V. CONCLUSIÓN

En este trabajo se ha propuesto un sistema para la mejora de las prestaciones de los NSM existentes en tres aspectos clave: la integración y parametrización de la información procedente de diversas fuentes heterogéneas, la selección automática de los incidentes más relevantes y la incorporación de información semántica al proceso de análisis. Cada una de estas contribuciones resulta relevante, ya que facilitan la tarea de los administradores de seguridad durante el proceso de monitorización y verificación de las alertas generadas por los sistemas automáticos, que pueden resultar muy numerosas y, consecuentemente, inmanejables.

La metodología propuesta ha mostrado una gran capacidad para dirigir al administrador hacia los incidentes relevantes y su interpretación. La evaluación realizada sobre el reto VAST12 ha permitido identificar todos los incidentes reportados hasta la actualidad en dicho reto, así como algunos que no habían sido detectados.

El sistema se encuentra actualmente implementado en laboratorio a nivel de realización de los análisis PCA y la obtención de las diferentes gráficas de forma no integrada, esto es, se requiere de la intervención del administrador en cada paso para ejecutar y proporcionar las entradas a cada módulo. Consecuentemente, una de las líneas de trabajo futuro debe centrarse en la integración de todas las herramientas en un NMS de fácil uso, automatizando el sistema y posibilitando el acceso a los datos originales a partir de los hallazgos del mismo para su inspección por parte del administrador.

AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el MICINN a través del proyecto TEC2011-22579.

REFERENCIAS

[1] R. Bejtlich, "The Tao of Network Security Monitoring", *Addison-Wesley*, 2004.
[2] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges", *Computers & Security*, 28(1):18–28, 2009.
[3] G. Keren, C. Lewis, "A Handbook for data analysis in the behavioral sciences: statistical issues,", *L. Erlbaum*, 1993.
[4] "Vast challenge 2012", http://www.vacommunity.org/vast+challenge+2012.
[5] I.T. Jollifee, "Principal component analysis", *Springer-Verlag*, 2002.
[6] P. Geladi, B.R. Kowalski, "Partial Least-Squares Regression: a tutorial", *Analytica Chimica Acta*, 185:1–17, 1986.
[7] J. Camacho, "Missing-data theory in the context of exploratory data analysis," *Chemometrics and Intelligent Laboratory Systems,* 103:8–18, 2010.
[8] J. Camacho, "Observation-based missing data methods for exploratory data analysis to unveil the connection between observations and variables in latent subspace models," *Journal of Chemometrics,* 25(11):592–600, 2011.
[9] J. Camacho, A. Ferrer,"Cross-validation in {PCA} models with the element-wise k-fold (ekf) algorithm: Practical aspects," *Chemometrics and Intelligent Laboratory Systems,* 131:37–50, 2014.
[10] J.E. Jackson, "A User's Guide to Principal Components," *Wiley*, 2003.
[11] G. Caraux and S. Pinloche, "Permutmatrix: a graphical environment to arrange gene expression profiles in optimal linear order," *Bioinformatics*, 21(7):1280–1, 2005.
[12] J. Camacho, "EDA toolbox", disponible en http://wdb.ugr.es/~josecamacho/downloads.php, 2013.
[13] R. Marty, "Applied Security Visualization," *Pearson Education*, 2008.
[14] F. Fischer, J. Fuchs, F. Mansmann, D. A. Keim, "Banksafe: A visual situational awareness tool for large-scale computer networks: Vast 2012 challenge award: Outstanding comprehensive submission, including multiple vizes," en *Proc. IEEE VAST*, pp. 257–258, IEEE Computer Society, 2012.
[15] Y. Cao, R. Moore, P. Mi, A. Endert, C. North, R. C. Marchany, "Dynamic analysis of large datasets with animated and correlated views: Vast 2012 mini challenge 2 award: Honorable mention for good use of coordinated displays.'", en *Proc. IEEE VAST*, pp. 283–284, IEEE Computer Society, 2012.
[16] L. Shi, Q. Liao, C. Yang, "Investigating network traffic through compressed graph visualization: Vast 2012 mini challenge 2 award: good adaptation of graph analysis techniques," en *Proc. IEEE VAST*, pp. 279–280, IEEE Computer Society, 2012.

Sistema de Detección de Anomalías para protocolos propietarios de Control Industrial

Iñaki Garitano[1], Mikel Iturbe, Ignacio Arenaza-Nuño, Roberto Uribeetxeberria, Urko Zurutuza

Dpto. de Electrónica e Informática
Escuela Politécnica Superior
Mondragon Unibertsitatea
Email: {igaritano,miturbe,iarenaza,ruribeetxeberria,uzurutuza}@mondragon.edu

***Resumen*—Las Infraestructuras Críticas, ofrecen servicios esenciales para el funcionamiento de sociedades modernas y se controlan mediante Sistemas de Control Industrial. Garantizar su seguridad es primordial debido a las graves consecuencias que puede acarrear un ataque exitoso. Además, la reciente aparición de gusanos diseñados de manera exclusiva evidencia el creciente interés que sufren dichos sistemas. Las soluciones de seguridad existentes se centran en protocolos de red públicos de Sistemas de Control Industrial, dejando a un lado los propietarios, debido en gran medida a su desconocimiento. Con el propósito de ofrecer un mecanismo de seguridad integral, tanto para protocolos propietarios como públicos, a lo largo de este artículo se presenta un Sistema de Detección de Anomalías basado en el *payload* y el flujo de los paquetes, en conjunto con un método capaz de describir el comportamiento de red mediante un conjunto de reglas. La validación se ha realizado utilizando un Sistema de Control Industrial real. El bajo número de falsos positivos demuestra su validez.**

***Palabras clave*—detección de anomalías (*anomaly detection*), protocolos propietarios (*proprietary protocols*), sistemas de control industrial (*Industrial Control Systems*)**

I. Introducción

Los Sistemas de Control Industrial (*Industrial Control Systems*, ICS) hacen referencia al conjunto de elementos especializados en la monitorización y control de procesos industriales, los cuales incluyen las Infraestructuras Críticas (*Critical Infrastructures*, CIs), necesarias para el correcto funcionamiento de las sociedades avanzadas.

Las Amenazas Persistentes Avanzadas (*Advanced Persistent Threats*, APTs) suponen una nueva generación de software malicioso y sofisticado. Con unas metas concretas y bien definidas, tienen un nivel de eficacia muy elevado y son sigilosas durante su ejecución, siendo capaces de ocultarse ante las posibles medidas de seguridad. Algunas de las APTs diseñadas para atacar a las CIs e ICSes son Stuxnet [1] y Duqu [2], cuyos objetivos son la interrupción de los servicios o el ciberespionaje y el robo de información.

Los ataques contra los ICSes, y en consecuencia la interrupción de sus servicios, podía acarrear serias consecuencias de diversa índole (económica, medioambiental…), potencialmente catastróficas, como muestran el impacto causado por algunos ataques anteriores [3]. Por ello, es de vital importancia salvaguardar la seguridad y el correcto funcionamiento de las CIs e ICSes ante las APTs y otros softwares maliciosos, ya que de ello depende en gran medida el bienestar de las sociedades avanzadas. Además, la seguridad es un proceso continuo donde cada medida de seguridad aporta o establece una barrera más, en un entorno donde no existe la seguridad absoluta. De ahí que sea necesario encontrar y desarrollar nuevas técnicas de seguridad que sean capaces de detectar no solo ataques bien conocidos, sino también posibles amenazas que pudiesen alterar el funcionamiento de los ICSes.

Los Sistemas de Detección de Intrusiones (*Intrusion Detection Systems*, IDS) se clasifican en base a distintos parámetros como bien pueden ser: el origen de los datos auditados, el método de detección o el modo de repuesta. Si bien es cierto que los IDSes se pueden catalogar en base a distintos atributos, en el entorno de los ICSes se clasifican principalmente en base al método de detección. Aquí podemos diferenciar dos grandes familias: las que se basan en el conocimiento, también conocidos como los basados en firmas, y los que se basan en el comportamiento, conocidos como Sistemas de Detección de Anomalías (*Anomaly Detection Systems*, ADS).

El método tradicional de detección de intrusiones está basado en firmas, donde las firmas describen patrones de ataque y el tráfico de red es analizado para ver si corresponde con alguna de las firmas. Esta estrategia sólo detecta ataques conocidos, debido a que es necesario conocer los detalles de un ataque para crear las firmas que lo describan. Por ello, los IDSes basados en el conocimiento son ineficientes a la hora de detectar APTs [4]–[5] y ataques desconocidos (*zero-day attacks*).

Las diferencias existentes entre los ICSes y las tradicionales Tecnologías de la Información y de las Comunicaciones, tales como los protocolos de red correspondientes a los ICSes o la falta de recursos de la mayoría de los componentes industriales, ponen de manifiesto la necesidad de la creación de medidas de seguridad especialmente diseñados para los ICSes [6]–[7]. Si bien es cierto que la seguridad de red se puede aportar a través de mecanismos de seguridad, tales como cortafuegos o IDSes, estos últimos son especialmente adecuados debido a las peculiaridades del tráfico de red industrial. En la mayoría de los casos, aun teniendo un ICS que controla un proceso físico continuo y cambiante, la comunicación entre

[1]Afiliación actual: University Graduate Center at Kjeller (UNIK), Noruega. Email: igaritano@unik.no

los *Master Terminal Unit* (MTU) o servidores de control y los *Remote Terminal Unit* (RTU) o *Programmable Logic Controller* (PLC), siguen patrones repetitivos y prácticamente estáticos [8]. Esta característica permite describir el tráfico de red a través de patrones de comportamiento para su posterior utilización por los IDSes.

Mientras que la mayoría de los ICSes utilizan protocolos públicos, también es cierto que un número importante de ICSes utilizan protocolos propietarios. Es decir, protocolos privados cuyas especificaciones se desconocen o sólo están disponible bajo acuerdos de confidencialidad. Actualmente, la mayoría de las soluciones y tecnologías de seguridad sólo están disponibles para protocolos públicos, lo cual pone de manifiesto la necesidad de crear herramientas de seguridad capaces de trabajar tanto con protocolos públicos como propietarios.

I-A. Contribución y organización del artículo

A lo largo de este artículo se presenta un método para la detección de anomalías de protocolos industriales tanto propietarios como públicos, que se transportan por encima del protocolo TCP/IP. La detección se realiza mediante el análisis de la carga útil o *payload* de los paquetes de red y la información de los flujos de la red. La sección II recopila el trabajo realizado en el campo de los detectores de anomalías que agrupan la carga útil de los paquetes o trabajan con protocolos cifrados. La sección III describe la arquitectura del sistema de detección de anomalías presentado. La sección IV muestra los resultados experimentales que miden el rendimiento del sistema presentado. Por último, las secciones V y VI extraen las conclusiones e identifican posibles líneas futuras de trabajo, respectivamente.

II. Trabajos relacionados

Existe una gran variedad de ADSes relacionados con los ICSes ([9]–[10]). La mayoría de los ADSes responden a protocolos públicos, es decir, protocolos cuyas especificaciones son conocidas. Esto permite conocer el propósito de los campos que componen cada paquete de red, lo cual hace posible saber *qué* está transportando la carga útil del paquete. Así el ADS será capaz de detectar las anomalías cada vez que el contenido se queda fuera del criterio preestablecido. Esta metodología es conocida como *Deep Packet Inspection* (DPI). Sin embargo, en el caso de protocolos propietarios la carga útil de las tramas de red es ininteligible, lo cual dificulta en gran medida el uso de la metodología DPI.

Aunque no directamente relacionados con protocolos propietarios, varios ADSes agrupan las cargas útiles de los paquetes para detectar anomalías. Estos sistemas están generalmente basados en n-gramas, cuya viabilidad para detectar anomalías en cualquier tipo de tráfico lo demostraron Bigham et al. [11], incluso en algunos casos en los que el tráfico de red está cifrado.

Anagram [12] es un detector de anomalías basado en análisis de n-gramas capaz de detectar ataques miméticos. Para este fin utiliza la randomización junto con filtros Bloom, reduciendo de esta manera la sobrecarga de cálculo. Aunque es una mejora de PAYL [13], sin embargo Anagram presenta algunas deficiencias y es susceptible a ataques como demostraron Pastrana et al. [14].

McPAD [15] utiliza una versión modificada de análisis basado en bigramas con el objetivo de detectar octetos correspondientes a *shellcodes*. Sin embargo, McPAD, en el caso de existir ligeras diferencias entre el conjunto de aprendizaje y un ataque, no es eficiente a la hora de detectar ataques.

Hadžiosmanović et al. [4] realizan una comparación de diferentes algoritmos basados en n-gramas para el análisis de anomalías en protocolos binarios entre los que se encuentra el protocolo de control industrial Modbus. Entre los algoritmos analizados, Anagram [12] es el que mejores resultados obtiene a la hora de detectar anomalías en las pruebas realizadas con el protocolo de control Modbus.

Otra aproximación a la detección de anomalías en tráfico desconocido es la realizada por Hoeve [16], el cual presenta una metodología para detectar intrusiones en tráfico de control cifrado. Para ello no inspeccióna la carga útil de los paquetes, sino que se basa en separar las inserciones de tráfico producidas por comandos, y reconocer las inserciones conocidas para luego alertar de las que no lo son. Sin embargo, a la hora de detectar anomalías en tráfico de control, es más deseable una granularidad de inspección alta [17] ya que es capaz de identificar ataques de inyección de datos.

III. Descripción del sistema

Entre los ADS mencionados en la sección II no hay ninguno que combine la información granulada de la carga útil junto con la información de flujo que posibilite la detección de anomalías en protocolos de control industrial. En esta sección presentamos una solución aplicable a protocolos propietarios y públicos, que utiliza la agrupación de los octetos de la carga útil, sin intentar interpretar su contenido, junto con la información de flujo para la detección de anomalías. Cabe mencionar que la idea principal del ADS presentado es crear un modelo de comportamiento para cada segmento de red entre los posibles RTUs y MTUs del sistema. Posteriormente cada modelo es sintetizado en un conjunto de reglas los cuales describen el comportamiento esperado, no los patrones de ataque.

La figura 1 muestra la estructura del ADS propuesto. El proceso de generación del patrón de comportamiento del tráfico de red y su sintetización en reglas se hace en modo fuera de línea, para luego utilizarlo en tiempo real. Así se establece si el tráfico de red se rige bajo unos limites preestablecidos. Seis componentes forman el ADS presentado, descritos a continuación:

Analizador de paquetes. Captura el tráfico de red y filtra los protocolos comunes tales como DNS o ARP dejando sólo el tráfico ICS relevante. Una vez obtenido y filtrado, guarda el tráfico capturado en un fichero binario con formato *pcap*.

Extractor de características. Recibe un fichero de captura como entrada y extrae todas las características necesarias para

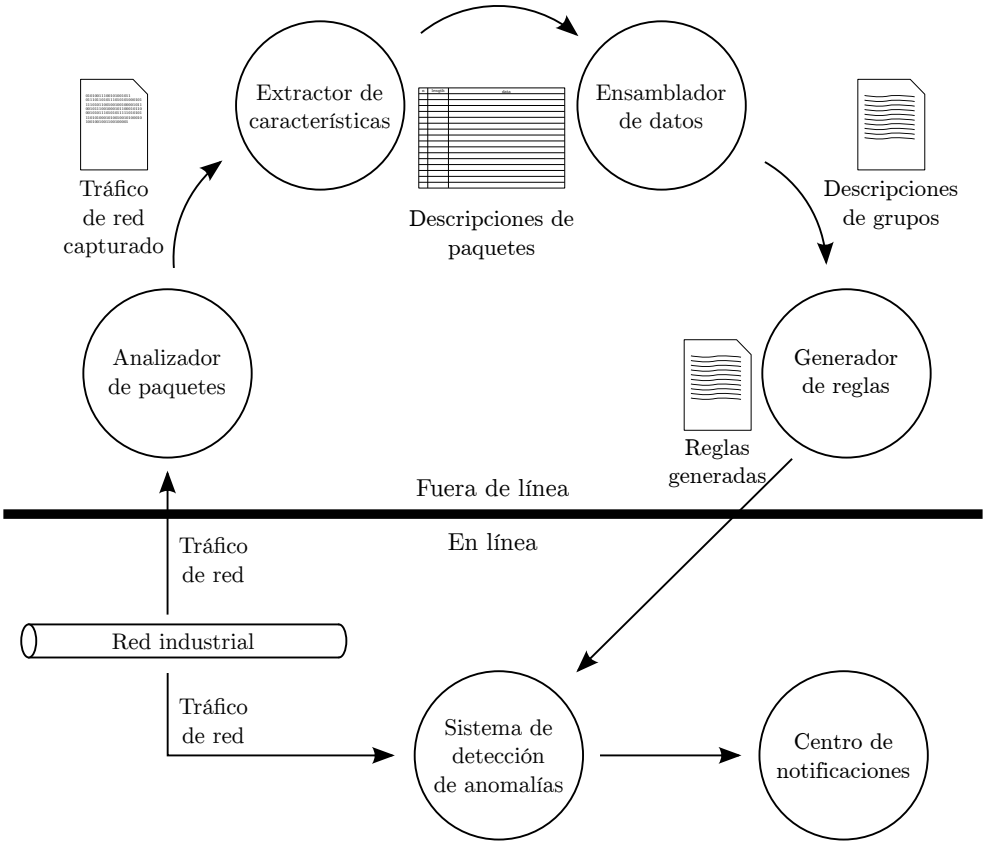


Figura 1. Componentes del Sistema de Detección de Anomalías.

crear el patrón de comportamiento que posteriormente será sintetizado en reglas. Estas son las características extraídas:

- Número de paquete
- *Timestamp*
- Dirección IP de origen y destino
- Número de puerto de origen y destino
- Longitud de la carga útil (número de octetos)
- Carga útil (cadena hexadecimal)

Ensamblador de datos. Es la pieza clave de todo el sistema y donde reside la lógica del método presentado. Su función se centra en analizar y comparar cada octeto de la carga útil de todos los paquetes que forman el mismo flujo de tráfico de red y en base a los criterios preestablecidos, formar grupos de octetos y extraer sus características. A continuación se detalla cada uno de los pasos que realiza el ensamblador de datos:

1. Ordena los paquetes según al flujo de red al que pertenecen, esto es, los clasifica en base a la dirección IP de origen y dirección IP de destino. Una vez ordenados, extrae el contenido por encima del protocolo TCP/IP, la carga útil, de cada uno de los paquetes.
2. Por cada flujo de red, organiza los octetos de la carga útil de los paquetes en filas y columnas. Cada fila contiene la carga útil de un único paquete, mientras que cada columna está formada por un único octeto, el octeto correspondiente a la posición de la columna.
3. Por cada columna compara todas las filas, esto es, compara la misma posición de octeto de todos los paquetes y así identifica las columnas (los octetos) cuyo valor cambia entre diferentes filas (paquetes).
4. Agrupa las columnas adyacentes cuyo valor cambia entre las distintas filas. Así se crean grupos de octetos cuya longitud, n, es el número de octetos cambiantes consecutivos. La tabla I muestra cinco grupos distintos, siendo tanto el grupo dos como el cinco del flujo 2 bigramas, esto es, grupos formados por dos octetos.
5. Identifica la posición del octeto inicial de cada grupo y anota tanto la posición como el número de octetos que lo constituyen.
6. Identifica y anota todos los posibles valores de cada grupo, es decir, los distintos valores de los octetos que forman cada grupo.
7. Una vez identificados y definidos todos los grupos, la información se envía al siguiente componente del ADS, el generador de reglas.

Generador de reglas. Este componente es el encargado de sintetizar el patrón de comportamiento, cada grupo y sus características, en un conjunto de reglas. Dichas reglas serán utilizadas posteriormente por el ADS, con el fin de comparar la situación en tiempo real con el patrón de comportamiento de red y así detectar las anomalías. Nótese que cada conjunto de reglas describe el tráfico de un único proceso industrial y un único segmento de red, por ello es necesario crear un

Tabla I
EJEMPLO DE LOS GRUPOS DE OCTETOS.

Núm. paquete	Carga útil (Octetos)													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	32	01	00	07	56	80	04	A0	00	00	40	61	20	...
2	32	01	00	07	66	80	05	60	00	00	40	71	20	...
7	32	01	00	07	76	80	04	A0	00	00	40	61	20	...

Flujo 1. Carga útil de los paquetes que forman el flujo dirección IP de origen 192.168.1.2 y dirección IP de destino 192.168.1.240.

Grupo 1 Grupo 2 Grupo 3

Núm. paquete	Carga útil (Octetos)													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	32	03	00	00	75	68	00	02	00	D0	00	00	04	...
4	32	03	00	00	76	68	00	02	00	90	00	00	04	...
5	32	07	00	00	00	00	00	0C	00	D6	00	01	12	...

Flujo 2. Carga útil de los paquetes que forman el flujo dirección IP origen 192.168.1.240 y dirección IP destino 192.168.1.2.

Grupo 1 Grupo 2 Grupo 3 Grupo 4 Grupo 5

nuevo conjunto de reglas para cada segmento de red o proceso industrial, además de cuando el proceso en cuestión sufre alteraciones. Las reglas generadas siguen el formato de Snort [18], un IDS basado en el tráfico de red flexible y ligero. Las reglas permiten a una versión modificada de Snort verificar el contenido de cada paquete de red para comprobar si la carga útil contiene alguna anomalía.

Las reglas de Snort se dividen en dos secciones lógicas: la cabecera y las opciones. El propósito de la cabecera es establecer la acción que Snort deberá realizar en el caso de detectar alguna anomalía e identificar cada flujo de paquetes. Para ello, en la cabecera se definen las siguientes opciones:

- Acción de la regla. Define qué acción realizar cuando se cumplen las opciones definidas en la regla.
- Protocolo de red.
- Dirección IP de uno de los nodos del flujo.
- Máscara de red de uno de los nodos del flujo.
- Número de puerto de uno de los nodos del flujo.
- Dirección del tráfico.
- Dirección IP de otro de los nodos del flujo.
- Máscara de red de otro de los nodos del flujo.
- Número de puerto de otro de los nodos del flujo.

En el caso de las opciones de la regla, el número de palabras clave disponibles es muy elevada, a continuación se detallan aquellas que son necesarias para este caso en concreto:

- Mensaje. Define el mensaje a mostrar junto con la alerta.
- Tamaño de la carga. En este caso se indica el tamaño mínimo.
- Contenido de la carga. Permite buscar contenido específico en la carga útil del paquete. Es este caso su cometido es detectar la ausencia de un valor en concreto.
- Posición del octeto inicial del patrón.
- Longitud del patrón (número de octetos).
- Flags del protocolo TCP.
- Número de identificación de la regla.
- Versión de la regla. Identifica inequívocamente revisiones de reglas.

La tabla II muestra las dos reglas generadas para los primeros dos grupos del flujo 1 mostrados en la tabla I. La primera regla, la cual corresponde al primer grupo, verifica si los paquetes cumplen los siguientes requisitos:

- Utiliza el protocolo TCP.
- La dirección IP de origen es 192.168.1.2.
- El puerto de origen es 102.
- Está dirigido a cualquier puerto de la IP 192.168.1.240.
- El tamaño de la carga útil es mayor a ocho octetos.
- El contenido del octavo octeto no corresponde a los valores `0x56`, `0x66` ó `0x76`.
- Tiene los flags ACK y PUSH de TCP.

Sistema de detección de anomalías. El objetivo de este componente es comparar en tiempo real el tráfico de red con las reglas generadas en modo fuera de línea y en el caso de detectar alguna inconsistencia alertar e identificar los paquetes. Para ello, se utiliza una versión modificada de Snort [18]. La razón de modificar Snort es que, por defecto, Snort no puede valerse de las reglas creadas por el módulo generador de reglas. La mayor limitación reside en que Snort no acepta reglas con un número de opciones mayor a 256. Esto limita en gran medida el tamaño y la cantidad de miembros de los grupos generados. Debido a ello, se ha procedido a la modificación de Snort, evitando así que el número de opciones máximo sea un impedimento a la hora de la ejecución.

Centro de notificaciones. Este componente es una interfaz de usuario gráfica que posibilita el análisis de las anomalías reportadas por Snort. Su arquitectura se basa en *Basic Analysis and Security Engine* (BASE) [19].

IV. RESULTADOS EXPERIMENTALES

La validación del sistema presentado se ha realizado mediante el uso de tráfico de red real de un ICS. Cabe mencionar la dificultad de la obtención de tráfico ICS real y que esté libre de ataque. Además, hasta donde sabemos, no existe ningún

Tabla II
EJEMPLO CON LAS REGLAS CREADAS PARTIENDO DE DOS GRUPOS DE LA TABLA I.

```
alert tcp 192.168.1.2 102 -> 192.168.1.240 any (msg:"plc_group1"; dsize:8<; \
content:!"|56|"; offset:8; depth:1; \
content:!"|66|"; offset:8; depth:1; \
content:!"|76|"; offset:8; depth:1; \
flags:PA; sid:1000001; rev:1;)

alert tcp 192.168.1.2 102 -> 192.168.1.240 any (msg:"plc_group2"; dsize:10<; \
content:!"|04A0|"; offset:10; depth:2; \
content:!"|0560|"; offset:10; depth:2; \
flags:PA; sid:1000002; rev:1;)
```

conjunto de datos estándar para realizar pruebas de este tipo. El ICS del cual se ha obtenido el tráfico de red controla un proceso de laminación de metal. El tráfico de red ha sido capturado en dos segmentos de red distintos, lo cual permite la evaluación de la respuesta del sistema ante tráfico diferente. Las dos capturas se han realizado en condiciones libres de ataques, en un entorno sin anomalías antes, durante y después de la captura. Los Controladores Lógicos Programables (*Programmable Logic Controller*, PLC) y conmutadores de red que componen la red del ICS corresponden a las familias Siemens S7 y Siemens SCALANCE, respectivamente.

La tabla III muestra los detalles de cada una de las capturas de tráfico de red realizadas para el proceso de generación de reglas y validación del sistema presentado. Las capturas tienen una duración de entre dos horas y de dos horas y media, lo cual se traslada a un total de 309.830 y 329.742 paquetes de tráfico de red respectivamente. Una vez filtrados los paquetes, las capturas se componen 99.808 y 138.855 paquetes correspondientes al protocolo Siemens S7.

Tabla III
CAPTURAS DE RED PARA LA GENERACIÓN DE REGLAS Y VALIDACIÓN DEL SISTEMA.

Núm. captura	Duración (segundos)	Núm. total de paquetes	Núm. de paquetes útiles
1	9013	309.830	99.808
2	7198	329.742	138.855

Para el proceso de generación de reglas que describen el patrón de comportamiento del tráfico de red, se ha utilizado el 75 % de cada captura de red. La tabla IV muestra el número de paquetes utilizado para la creación del patrón de comportamiento, esto es, la creación de los grupos, así como el número de grupos generados para cada captura y su longitud máxima y mínima en número de octetos. Hay que tener en cuenta que cada grupo se sintetiza en una única regla. Una vez generadas las reglas fuera de línea, cada fichero de captura al completo ha sido comprobado con su conjunto de reglas correspondiente. El ADS ha verificado todo el tráfico, alertando al centro de notificaciónes de cada una de las anomalías detectadas.

La tabla V muestra los resultados obtenidos. En ambos casos, se puede observar que el número de falsos positivos respecto al 25 % de los paquetes no utilizados en el proceso de generación de reglas es relativamente bajo, lo cual indica

Tabla IV
CARACTERÍSTICAS DE LOS PATRONES DE COMPORTAMIENTO GENERADOS.

Núm. captura	Núm. de paquetes	Núm. de grupos	Longitud mín. (octetos)	Longitud máx. (octetos)
1	74.856	52	1	71
2	104.141	6	1	227

que el conjunto de reglas creado tiene una exactitud muy alta. Si se analiza toda la captura de red, no solo el restante 25 % de los paquetes, sino el total de la captura, el número de falsos positivos se mantiene. Esto indica que el proceso de generación de reglas es capaz de sintetizar en reglas todas las posibles opciones y que el ADS no genera más falsos positivos que lo necesario. Obviamente, el porcentaje disminuye al comparar el número de falsos positivos con el total de los paquetes capturados (0,016 % y 0,052 % respectivamente). En cuanto al número de reglas activadas a causa de alguna anomalía detectada, en el caso de la primera captura sólo siete de ellas han generado una alerta. Eso significa que los 49 paquetes que han registrado un falso positivo, están concentrados en esas siete reglas. En cuanto a la segunda captura, la concentración de reglas activadas es todavía mayor, ya que pese a que hay un número mayor de paquetes falsos positivos, el número de reglas afectadas es menor. Sin embargo, porcentualmente, el número de reglas que generan falsos positivos es mayor en la segunda captura que en la primera.

Tabla V
RESULTADOS EXPERIMENTALES: ANOMALÍAS DETECTADAS.

	Núm. de captura	
	1	2
Paquetes falsos positivos (F.P.)	49	173
F.P. vs. restante 25 % de paquetes	0,2 %	0,5 %
F.P. vs. todos los paquetes	0,016 %	0,052 %
Reglas falsas positivas	7	2
F.P. vs. todas las reglas	13,46 %	33,33 %

El número de falsos positivos obtenido se debe en gran medida a la existencia de un identificador de paquete en el protocolo transportado por encima del protocolo TCP/IP. Generalmente, los protocolos de red disponen de un identificador de paquete, esto es, un número que identifica cada paquete de manera inequívoca. Este número se incrementa de manera gradual hasta llegar a un límite establecido por

el número de octetos destinados para su representación. Así, si las capturas de tráfico incluyeran todos los valores posibles del identificador de paquete, el número de falsos positivos se reduciría de manera considerable. Con lo cual, el hecho de aumentar el tiempo de captura o el porcentaje de la captura utilizada para la generación de las reglas, reduciría el número de alertas. Otra solución sería la de descartar la regla que verifica la sección del identificador de paquete, entendiendo que este campo puede contener todos los valores posibles. No aporta valor verificar un campo cuyo contenido puede ser cualquiera. Además, esta solución reduciría la carga de trabajo del ADS. Otro de los resultados a resaltar es el hecho de que el sistema ha sido capaz de detectar todos aquellos paquetes cuyo contenido se ha modificado de manera manual.

V. Conclusiones

El creciente número de amenazas y el interés que despiertan los sistemas de control industrial hace necesaria su protección mediante la creación de nuevos sistemas de seguridad. En este artículo presentamos un sistema de detección de anomalías para protocolos propietarios de ICS basado en la agrupación de los octetos de la carga útil que además utiliza información de flujo de red. Los protocolos propietarios dificultan la creación de ADSes debido a la falta de las especificaciones del protocolo. Sin embargo, la carga útil del protocolo puede ser tratada como un conjunto de datos en crudo del cual se puede extraer un patrón de comportamiento habitual del sistema. La naturaleza repetitiva y estática del tráfico entre los MTUs y RTUs de los ICS hace posible que se puedan detectar anomalías, con un margen de falsos positivos relativamente bajo.

El sistema presentado consta de seis componentes, los cuales generan el patrón de comportamiento del tráfico de red, lo sintetizan en un conjunto de reglas y posteriormente, a través de un IDS modificado, detectan paquetes de red que se desvían del comportamiento habitual del sistema. La metodología descrita ha sido validada mediante dos capturas reales de tráfico de ICS, con un porcentaje de falsos positivos por debajo del 0,5 %. El método es extensible a diferentes protocolos de red de control industrial, tanto propietarios como públicos, ya que el sistema se abstrae del significado de los diferentes paquetes, centrándose sólo en el contenido.

VI. Trabajos futuros

A lo largo de este artículo, el tráfico se clasifica en base a las direcciones IP de origen y destino. La utilización de más características, como el tamaño del paquete, mejoraría el resultado del sistema de detección de anomalías. Generalmente los paquetes del mismo protocolo de un mismo tamaño tienen el mismo propósito (p. ej. los paquetes generados que actualizan el valor de una temperatura tendrán cargas útiles de la misma longitud).

Por otro lado, a lo largo de este trabajo solo se han contemplado aquellos octetos de la carga útil que cambian de valor a lo largo de distintos paquetes. La creación de grupos estáticos, es decir, secuencias de octetos idénticas a lo largo de todos los paquetes pueden ayudar a una representación más exacta del tráfico de red. Esto haría más difícil enmascarar un ataque, ya que el atacante no sólo debería replicar el contenido dinámico de los paquetes, sino también el estático, dificultando así la inserción de código malicioso.

Referencias

[1] N. Falliere, L. O. Murchu, and E. Chien, "W32.Stuxnet dossier," *White paper, Symantec Corp., Security Response*, 2011.

[2] B. Bencsáth, G. Pék, L. Buttyán, and M. Félegyházi, "Duqu: Analysis, detection, and lessons learned," in *ACM European Workshop on System Security (EuroSec)*, 2012.

[3] B. Miller and D. Rowe, "A survey SCADA of and critical infrastructure incidents," in *Proceedings of the 1st Annual conference on Research in information technology*, pp. 51–56, ACM, 2012.

[4] D. Hadžiosmanović, L. Simionato, D. Bolzoni, E. Zambon, and S. Etalle, "N-gram against the machine: On the feasibility of the n-gram network analysis for binary protocols," in *Research in Attacks, Intrusions, and Defenses*, pp. 354–373, Springer, 2012.

[5] C. Tankard, "Advanced Persistent threats and how to monitor and deter them," *Network security*, vol. 2011, pp. 16–19, August 2011.

[6] Keith Stouffer, Joe Falco, and Karen Scarfone, "Guide to Industrial Control Systems (ICS) Security, Special publication 800-82," tech. rep., National Institute of Standards and Technology, June 2011.

[7] B. Galloway and G. Hancke, "Introduction to Industrial Control Networks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 860–880, 2012.

[8] R. Chandia, J. Gonzalez, T. Kilpatrick, M. Papa, and S. Shenoi, "Security Strategies for SCADA Networks," in *Critical Infrastructure Protection* (E. Goetz and S. Shenoi, eds.), vol. 253 of *IFIP International Federation for Information Processing*, pp. 117–131, Springer US, 2008.

[9] I. Garitano, R. Uribeetxeberria, and U. Zurutuza, "A review of SCADA anomaly detection systems," in *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*, pp. 357–366, Springer, 2011.

[10] B. Zhu and S. Sastry, "SCADA-specific intrusion detection/prevention systems: a survey and taxonomy," in *Proceedings of the 1st Workshop on Secure Control Systems (SCS)*, 2010.

[11] J. Bigham, D. Gamez, and N. Lu, "Safeguarding SCADA Systems with Anomaly Detection," in *Computer Network Security* (V. Gorodetsky, L. Popyack, and V. Skormin, eds.), vol. 2776 of *Lecture Notes in Computer Science*, pp. 171–182, Springer Berlin Heidelberg, 2003.

[12] K. Wang, J. J. Parekh, and S. J. Stolfo, "Anagram: A content anomaly detector resistant to mimicry attack," in *Recent Advances in Intrusion Detection*, pp. 226–248, Springer, 2006.

[13] K. Wang and S. J. Stolfo, "Anomalous payload-based network intrusion detection," in *Recent Advances in Intrusion Detection*, pp. 203–222, Springer, 2004.

[14] S. Pastrana, A. Orfila, J. E. Tapiador, and P. Peris Lopez, "Randomized Anagram revisited," *Journal of Network and Computer Applications*, 2014.

[15] R. Perdisci, D. Ariu, P. Fogla, G. Giacinto, and W. Lee, "McPAD: A multiple classifier system for accurate payload-based anomaly detection," *Computer Networks*, vol. 53, no. 6, pp. 864 – 881, 2009. Traffic Classification and Its Applications to Modern Networks.

[16] M. Hoeve, "Detecting Intrusions in Encrypted Control Traffic," in *Proceedings of the First ACM Workshop on Smart Energy Grid Security*, SEGS '13, (New York, NY, USA), pp. 23–28, ACM, 2013.

[17] D. Hadžiosmanović, D. Bolzoni, S. Etalle, and P. Hartel, "Challenges and opportunities in securing industrial control systems," in *Complexity in Engineering (COMPENG), 2012*, pp. 1–6, June 2012.

[18] M. Roesch, "Snort – Lightweight Intrusion Detection for Networks," in *Proceedings of LISA '99: 13th Systems Administration Conference* (USENIX, ed.), vol. 99, (Seattle, WA, USA), pp. 229–238, November 1999.

[19] A. Flores, J. Fields, A. Graham, J. Hart, K. Johnson, D. Mackie, S. Muller, T. Rupp, C. Svensson, and M. Valdez, "Basic Analysis and Security Engine," 2010. Online. Accedido el 14 de Marzo de 2014 http://base.professionallyevil.com.

Protocolo para la Notificación y Alerta de Eventos de Seguridad en Redes Ad-hoc

Leovigildo Sánchez-Casado, Roberto Magán-Carrión, Pablo Garrido-Sánchez, Pedro García-Teodoro
Departamento de Teoría de la Señal, Telemática y Comunicaciones
Universidad de Granada - CITIC
Email: sancale@ugr.es, rmagan@ugr.es, pablogs9@correo.ugr.es, pgteodor@ugr.es

***Resumen*—Las líneas de defensa de seguridad tradicionales para proteger un sistema dado son prevención, detección y respuesta. A pesar de que sobre el papel dichos módulos deben inter-operar a fin de conseguir una seguridad integral, por lo general se plantean y adoptan como soluciones independientes. El presente trabajo aborda el estudio y desarrollo de un protocolo de notificación y alerta de eventos de seguridad cuyo fin principal es servir de interfaz entre los módulos de detección y respuesta.**

Ideado específicamente para redes ad-hoc, su uso posibilita poner en conocimiento de los elementos constitutivos del entorno monitorizado la ocurrencia de un cierto comportamiento malicioso detectado. Este conocimiento será clave para la ejecución posterior de los mecanismos de respuesta oportunos.

También susceptible de ser usada para la distribución de información en procesos de detección/respuesta colaborativos, nuestra propuesta viene a cubrir una carencia manifiesta en el campo objeto de estudio.

***Palabras clave*—seguridad en redes (*network security*), redes ad-hoc (*ad-hoc networks*), intrusión (*intrusion*), detección (*detection*), respuesta (*response*), notificación y alerta (*notification and alert*)**

I. Introducción

Las redes ad-hoc constituyen en la actualidad un paradigma de comunicaciones de uso creciente. La ausencia de infraestructura y la posible adopción de rutas origen-destino multi-salto hacen este tipo de entornos altamente atractivos para aplicaciones de carácter medioambiental, militar, gestión de situaciones de crisis (p.e., terremotos, atentados terroristas), etc. de alta autonomía [1]. Esta versatilidad se ve incrementada cuando, además, los nodos que conforman la red tienen capacidad de movilidad, lo que constituye las denominadas MANET (*Mobile Ad-hoc NETworks*) [2], nombre del que derivan y con el que están relacionados otros también conocidos como VANET (*Vehicular Ad-hoc NETworks*) y FANET (*Flying Ad-hoc NETworks*).

No obstante las ventajas de este tipo de entornos, son varias también sus limitaciones. Por una parte, por restricciones usuales relacionadas con el tiempo de vida de la batería, la capacidad de almacenamiento y la potencia de cómputo de los nodos. Por otro lado, y no menos importante que lo anterior, por los enormes riesgos de seguridad inherentes a este tipo de redes y sistemas [3]. Especialmente motivados por su naturaleza abierta (inalámbrica), además de la potencial inexistencia de una infraestructura de control y gestión centralizados, son varios los tipos de amenazas existentes [4]. Por mencionar algunos, sírvase citar, entre otros, ataques de *dropping*, en los que un nodo malicioso elimina paquetes en su ruta (multi-salto) hacia un destino dado; ataques de *jamming*, donde un nodo genera interferencias y evita el acceso con éxito de otros al canal de comunicaciones; ataques de suplantación de identidad (*spoofing, sybil*), consistentes en la falsificación de la identidad de un nodo; ataques de *route poisoning*, donde se falsifican las tablas de encaminamiento de los nodos a fin de atraer tráfico hacia una cierta zona con fines maliciosos (eliminación, falsificación, acceso no permitido).

Si bien son conocidas las medidas preventivas de adopción aconsejada para evitar la ocurrencia de actuaciones maliciosas como las antes referidas, como sucede en cualquier otro entorno de comunicaciones, su despliegue no garantiza en modo alguno la no aparición de las mismas. En consecuencia, ante la potencial superación de las barreras de seguridad preventiva dispuestas (generalmente basadas en el empleo de esquemas criptográficos), se precisa complementar éstas con otras orientadas a la detección de eventos indeseados. En este caso, sobre la base de la monitorización de la actividad habida en el entorno, el objetivo es determinar la ocurrencia de comportamientos maliciosos contra la seguridad del sistema. En respuesta a estos eventos debieran ser adoptadas las medidas oportunas para su resolución y, en suma, la recuperación del sistema. Adicionalmente, es recomendable la realimentación de todo el proceso a fin de permitir la adaptación dinámica del entorno (véase Figura 1) [5].

En la literatura se encuentran desarrollados numerosos esquemas de prevención, detección y respuesta (menos de los terceros que de los dos primeros), pero se evidencia una alta carencia de propuestas orientadas a la inter-operación de estos módulos. De esta manera, las soluciones de seguridad habitualmente disponibles son parciales por cuanto que sólo se enfocan en uno de los tres aspectos citados y, sobre todo, porque se desarrollan obviando la necesidad de disponer de procedimientos efectivos de comunicación entre los distintos

Figura 1: Líneas de defensa tradicionales ante incidentes de seguridad.

módulos. Esto es especialmente cierto y crítico para entornos de red ad-hoc [6]. En este caso, y frente a otros mecanismos existentes en la actualidad como es el simple envío de un correo electrónico al administrador del sistema, debe habilitarse algún procedimiento que permita la constatación de este hecho por parte del resto de la red para la subsiguiente ejecución de esquemas de respuesta aislados y/o distribuidos globalmente coherentes.

El presente trabajo aborda el diseño y uso de un protocolo de notificación y alerta de eventos de seguridad en la línea antes apuntada. Para ello, el resto del documento se organiza como sigue. En la Sección II se discuten algunas propuestas en la línea aquí planteada existentes en la bibliografía especializada. Tras ello, y habida cuenta de la baja idoneidad de las mismas para entornos ad-hoc, en la Sección III se presenta nuestra propuesta concreta y se discute su uso con varios fines relacionados. Seguidamente, la Sección IV se dedica a un breve análisis de prestaciones del protocolo introducido desde el punto de vista del impacto que tiene su uso sobre las comunicaciones del entorno. Finalmente, en la Sección V se concluye con los aspectos más relevantes de la propuesta realizada y se apuntan brevemente algunas actuaciones de futuro.

II. Trabajo relacionado

Podemos encontrar algunas propuestas de esquemas de notificación de incidentes de seguridad en la literatura. En concreto, es de reseñar el protocol IDXP [7] y el formato de mensajes asociado IDMEF [8] desarrollados por la IETF (http://www.ietf.org). Ambos están centrados exclusivamente en el manejo de información propia de un IDS (*Intrusion Detection System*), no siendo adecuados para datos relacionados con respuesta a incidentes en un contexto más general [9].

Otra propuesta de notificación es IODEF [10]. De tipo XML, IODEF no ha sido adoptado de forma masiva debido a los requerimientos en cuanto a las herramientas necesarias para soportarlo. Frente a este formato, es de mencionar la disponiblidad de otros protocolos y formatos de mensaje tales como X-ARF [11] y XMPP [12], [13], [14].

Sea como fuere, la escasa generalización alcanzada por las propuestas mencionadas hace que las recomendaciones acerca de soluciones de gestión de información de incidentes de seguridad tiendan hacia el uso de ficheros de tipo texto, aconsejándose el empleo de formatos ligeros como CSV (*Comma Separated Value*).

Al margen de las propuestas específicas para incidentes de seguridad antes comentadas, otra posibilidad para el manejo de información relacionada con una red es *syslog* [15]. Desarrollado en la década de 1980 como parte del proyecto Sendmail para tracear los eventos de un sistema, *syslog* permite la separación del software que genera los mensajes del sistema que los almacena y del software que los analiza. Los mensajes *syslog* están etiquetados con un código indicativo del tipo de software que los generó (ftp, mail, etc.) y un grado de severidad (desde *Emergency*, el más alto, hasta *Debug*, el más bajo). Aunque *syslog* puede utilizarse para la gestión de eventos de seguridad, su complejidad (derivada de su amplia versatilidad) hace que este estándar no resulte el mejor candidato para el fin que aquí perseguimos. En especial para redes ad-hoc, donde, según lo ya apuntado en la Sección I, interesaría la adopción de soluciones específicas y, en consecuencia, ligeras desde el punto de vista del coste y carga implicados.

Seguidamente se describe la propuesta de notificación y alerta en nuestro caso adoptada. Ésta, frente a las anteriores, está específicamente diseñada para su uso en entornos ad-hoc, de manera que resulte los menos costosa posible desde el punto de vista de los recursos requeridos.

III. Protocolo de notificación y alerta de eventos de seguridad

Como ya se ha comentado anteriormente, no existen reportadas en la literatura soluciones adecuadas para la notificación de eventos de seguridad en redes ad-hoc. Ideada tomando como base el protocolo de *routing* AODV [17] para este tipo de redes, la propuesta particular que en este apartado se desarrolla presenta las siguientes características principales:

- Versátil, al implementarse sobre la capa de aplicación (concretamente sobre el puerto 703, actualmente sin asignación).
- Rápido y eficiente, definiéndose sobre UDP para reducir retardos y consumo de recursos.
- Flexible, ya que posibilita su uso con diversos fines, contemplándose en la versión actual dos principales:
  - notificación de eventos de seguridad una vez que se haya detectado la ocurrencia de incidentes reseñables, e
  - intercambio de información orientada a la potencial detección colaborativa de tales eventos o a la respuesta ante los mismos.
- El envío de estos mensajes se prevé en tres variantes: *unicast*, *broadcast* a toda la red y *broadcast* a los vecinos, es decir a un salto (TTL=1), dependiendo del tipo de mensaje concreto de que se trate.

Es evidente la necesidad de definir los mensajes específicos que darán soporte a las funcionalidades mencionadas, así como los aspectos relacionados con el envío de los mismos. Seguidamente se discute en detalle todo ello.

III-A. Usos y tipos de mensajes

Como se ha comentado desde el principio, el protocolo está inicialmente pensado para llevar a cabo la notificación de alertas de seguridad ante la constatación de ciertos incidentes en el entorno ad-hoc monitorizado. Esta comunicación permitirá, en su caso, el despliegue posterior de medidas de respuesta orientadas a dar solución a los incidentes reportados. No obstante este fin principal, también es posible la adopción del protocolo para otros objetivos no menos interesantes en el contexto de la seguridad que nos ocupa. Para ello se propone un diseño flexible a través de la especificación de diversos tipos de mensajes. En concreto, en este punto se plantea un segundo uso del protocolo: intercambio de información

de seguridad entre los nodos del entorno para, por ejemplo, posibilitar una detección de eventos maliciosos de forma colaborativa o una respuesta coordinada frente a los mismos.

III-A1. Notificación de alertas: Como ya se ha indicado con anterioridad, es manifiesta la ausencia de procedimientos de alerta de eventos de seguridad en entornos de red; en particular, para redes ad-hoc. Tomando como base los desarrollos IDS realizados por los autores [18], [19], al tiempo que la experiencia en esquemas de respuesta [20], [21], se plantea un procedimiento de notificación de alertas de seguridad para la comunicación de la siguiente información una vez determinada la ocurrencia de un evento intrusivo malicioso:

- *Tipo de mensaje*: necesario para diferenciar entre los distintos usos ya apuntados para el protocolo pretendido.
- *Tipo de evento detectado*: teniendo presentes las distintas tipologías existentes (*dropping*, *sinkhole*, etc. [4]), parece evidente que los posibles mecanismos de respuesta a desplegar dependerán de la tipología concreta del ataque.
- *Severidad del evento*: para indicar el grado de afectación de éste. Por ejemplo, no es lo mismo un ataque de *dropping* donde se descarte un 20 % de los paquetes a retransmitir que uno donde se descarten todos ellos.
- *Confiabilidad de la detección*: para indicar el grado de certeza con el que se concluye el proceso de detección. Asi, por ejemplo, no es comparable la detección de un ataque fundamentada en la observación de un patrón conocido (basada en firmas o *misuse*) que una derivada de la desviación del comportamiento del sistema analizado (detección basada en anomalías). Es evidente que en el primer caso la confiabilidad será del 100 %, mientras que en el segundo será función (previsiblemente) del grado de desviación observado [22].
- *Identidad del nodo malicioso*: necesaria para la adopción de ciertos mecanismos de respuesta específicos (p.e., el aislamiento del nodo en cuestión). Esta identidad se refiere típicamente a la dirección IP del nodo atacante.
- *Identidad del nodo detector*: similar a la anterior, ésta identifica el nodo que detectó el incidente reportado y que corresponde con el nodo emisor del mensaje de notificación.
- *Instante de detección*: identificativo del momento temporal en el que se produjo la observación del incidente de seguridad reportado. Esta información puede resultar útil de cara a la correlación de eventos.

Adicionalmente a la información principal anterior, centrada en el evento de seguridad específico detectado, otra información oportuna a considerar en los mensajes es:

- *Identificador del mensaje*: como es habitual en numerosos protocolos de comunicaciones, este valor se refiere a un número monótonamente creciente identificativo del mensaje para, entre otros fines, robustecer el protocolo ante ataques de repetición.
- *Longitud total*: debido principalmente al campo que sigue abajo, es preciso la indicación expresa de la longitud total (en palabras de 32 bits) del mensaje.

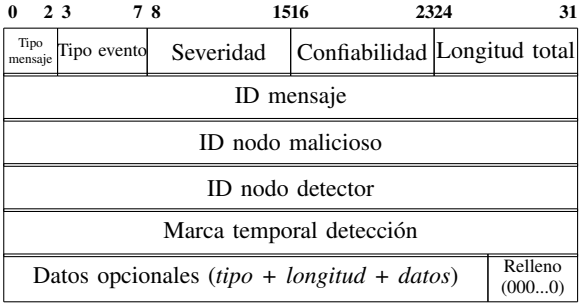


Figura 2: Formato de mensajes de notificación de alertas.

- *Datos (opcional)*: aunque en la versión actual no está definido, sería interesante la inclusión de otra posible información útil varia. Por ejemplo, la localización exacta del nodo malicioso para solucionar ataques de *jamming*. Sean cuales fueren estos posibles usos futuros, el formato de este campo debe ser:
  $< tipo\_datos > \ < longitud\_octetos\_datos > \ < datos >$
  Gracias al campo de *longitud total* previo referido, resulta posible el uso secuenciado de información extra diversa. También es de señalar la necesidad de, con objeto de que el mensaje sea múltiplo de 32 bits, contemplar un campo de *relleno* (*padding*) consistente en todo ceros, localizado (en su caso) al final del campo de información opcional.

De acuerdo con todo lo anterior, el formato de los mensajes de notificación de alertas de incidentes de seguridad propuesto es el mostrado en la Figura 2. En ella se indican los campos anteriormente referidos, junto con los bits asignados a cada uno ellos. Es de significar que algunos de los campos se proponen con una longitud superior a la estrictamente necesaria en este punto para posibilitar la expansión futura del protocolo.

III-A2. Intercambio de información de seguridad: Más allá de la indudable utilidad de los mensajes de alerta descritos, es manifiesto el posible uso del protocolo de notificación ideado para otros fines. Es el caso del potencial intercambio de información de seguridad entre nodos. Esta aplicación, al margen de la evidente similitud con esquemas como IDMEF o *syslog*, surge principalmente de los trabajos [18], [19], donde se plantean esquemas IDS colaborativos fundamentados en el intercambio de información entre nodos (principalmente vecinos). Éstos, frente a los de naturaleza aislada, donde cada nodo implementa su propio IDS a partir de información adquirida exclusivamente de forma local, persiguen la adopción de decisiones de detección más globales y, como tales, más robustas y fiables. Huelga decir que la aplicabilidad del citado intercambio incluye también IDS centralizados donde se precisa la adquisición de información de toda la red por parte de un solo nodo central. En uno y otro caso, centralizado y distribuido, el esquema de intercambio es totalmente análogo: existe un nodo (que implementa un IDS) que solicita información de otro cierto nodo (por ejemplo, porque el IDS local del solicitante ha disparado una alarma para él) a otros nodos de la red (todos, su vecindad, etc.), en respuesta a lo cual se proporciona la información específica solicitada para

facilitar la posterior decisión de detección.

Habida cuenta que los esquemas IDS propuestos por los autores son multi-capa (acceso al canal, capa de red, etc.), la información requerida se va a identificar en los mensajes intercambiados organizada en base a los procedimientos/protocolos específicos a los que aquélla se refiere. En la Figura 3 se muestra el formato específico de los mensajes involucrados en el intercambio de información. Por lo que respecta a los de solicitud (subfigura 3(a)), los campos involucrados son:

- *Tipo de mensaje*: a través del cual se indica la semántica del paquete. Solicitud de información de seguridad en este caso, frente a la notificación de eventos descrita en III-A1 o la respuesta a la solicitud descrita más adelante.
- *Número de entradas*: relativo a la cantidad de variables cuyo valor se solicita. Como se indica más adelante, cada variable se especifica a través de 32 bits.
- *Longitud total*: con el cual se especifica la longitud total (en octetos) del mensaje enviado.
- *ID mensaje*: como en los mensajes de notificación de alertas, si bien en este caso también se utilizará este campo para hacer corresponder solicitudes con respuestas.
- *ID nodo solicitado*: para identificar unívocamente el nodo del que se solicita la información.
- *ID nodo solicitante*: para identificar unívocamente el nodo que solicita la información y que, en definitiva, se prevé llevará a cabo el proceso de detección posterior.
- *Variable 1...n*: campos sucesivos de 32 bits de longitud a través de los cuales se identifica cada una de las *n* variables indicadas en el campo *número de entradas* de las que se pide información para el nodo solicitado. Como hemos mencionado anteriormente, cada variable queda definida (como se hace, por ejemplo, en las MIB de gestión) a partir de dos campos: *protocolo/procedimiento* al que hace referencia (por ejemplo, IP, ICMP, 802.11, etc.) e *identificador* dentro del mismo.

Por lo que respecta a los mensajes de respuesta a los de solicitud anteriores, su formato es el especificado en la Figura 3(b):

- *Tipo de mensaje*: respuesta a solicitud de información.
- *Número de entradas*: cantidad de variables cuyo valor se indica en el mensaje. Como se indica más adelante, cada variable implica el uso de 64 bits, 32 para su identificación y 32 para su valor.
- *Longitud total*: con el cual se especifica la longitud total (en octetos) del mensaje enviado.
- *ID mensaje*: para hacer corresponder solicitudes con respuestas.
- *ID nodo informado*: para identificar unívocamente el nodo del que se comunica la información.
- *ID nodo emisor*: para identificar unívocamente el nodo que envía la información.
- *Variable 1...n*: campos sucesivos de 64 bits de longitud a través de los cuales se identifica e informa de cada una de las *n* variables indicadas en el campo *número de entradas* de las que se requirió información para el *nodo solicitado* en el mensaje de petición (=*nodo informado*). Tras ser identificada cada variable (con 32 bits como se ha establecido antes, 8 de los cuales son para indicar el protocolo/procedimiento al que se refiere), seguidamente se especificará su valor mediante un campo de 32 bits.

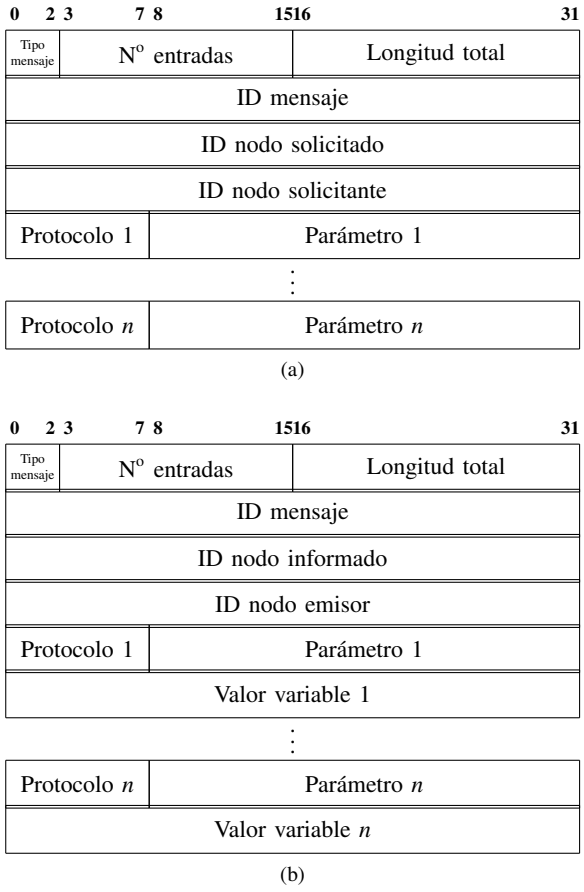


Figura 3: Mensajes de solicitud (a) y respuesta (b) de información.

Aunque la funcionalidad de intercambio de información objeto de estudio no se ha desarrollado completamente en la práctica, en la Tabla I se indican algunas de las variables consideradas y que son utilizadas en los IDS desplegados hasta la fecha por los autores, además de ser de amplio uso para este fin en la literatura. Así, el protocolo propuesto proporciona una gran flexibilidad, posibilitando la extensión del mismo mediante la definición e inclusión de nuevas variables.

No queremos concluir la exposición de los posibles usos y tipos de mensajes asociados al protocolo de notificación desarrollado sin reseñar de nuevo la versatilidad pretendida para el mismo. Así, por ejemplo, se podría definir un nuevo tipo de mensaje de *intercambio de información asíncrona* donde no se precise una solicitud previa. Para ello, por ejemplo, podríamos utilizar el mismo formato de la Figura 3(b) con los siguientes matices:

- *Tipo de mensaje*: fijado a un valor diferente de los tres previos ya descritos.
- *ID mensaje*: identificativo del paquete en sí y no para

Tabla I: Ejemplo de variables para intercambio de información.

Protocolo/ Procedimiento	Parámetro	Notas
Miscelánea	Período muestreo	En s
Topología	Velocidad	En m/s
	Aceleración	En m/s^2
	Localización	Posición GPS
Física	RSSI	De 0 a -80 dBm
MAC 802.11	$\#\mathrm{P}_{\mathrm{RTS}}$	Paquetes RTS / CTS enviados y recibidos
	$\#\mathrm{P}_{\mathrm{CTS}}$	
AODV	$\#\mathrm{P}_{\mathrm{HELLO}}$	Paquetes HELLO / RREQ / RREP enviados, recibidos, retransmitidos y descartados
	$\#\mathrm{P}_{\mathrm{RREQ}}$	
	$\#\mathrm{P}_{\mathrm{RREP}}$	
	NumSeq	Número de secuencia
	HopCount	Número de saltos
Aplicación	$\#\mathrm{P}_{\mathrm{datos}}$	Paquetes de datos enviados, recibidos y perdidos
	#Sesiones	Número de sesiones

hacer corresponder solicitudes con respuestas.

III-B. Distribución de mensajes

Un aspecto importante en el diseño de todo procedimiento de notificación de alertas es el esquema a emplear para la transmisión o envío de la información correspondiente, pues el objetivo es que los recursos implicados, y con ello el impacto sobre las comunicaciones globales, sean los menores posibles. Distintas posibilidades son contempladas para ello en la bibliografía: inundaciones, encaminamiento selectivo, agrupamiento, publicación/suscripción [16]. En nuestro caso, vamos a considerar las siguientes posibilidades en función de la aplicación y tipo de mensaje:

- *Broadcast* a toda la red para la notificación de alertas ante la detección de incidentes. Los nodos que reciban dichos mensajes podrán, a su vez, retransmitir la eventualidad reportada para su distribución a toda la red.
- *Broadcast* a los vecinos para los mensajes de solicitud de información. Para ello, estos paquetes serán enviados sobre la red con el campo TTL del paquete IP sobre el que se encapsulan a valor 1. Es de significar que este tipo de transmisión es más eficiente que el anterior por cuanto que permitiría la inclusión de inteligencia en las retransmisiones de los nodos para evitar informaciones duplicadas.
  También podría considerarse el envío *unicast*, dependiendo del deseo del nodo emisor en cuanto a información pretendida y procedencia de la misma.
- *Unicast* para los mensajes de respuesta hacia el nodo solicitante.

Todas estas cuestiones, así como la propia especificación de los mensajes, tiene un impacto directo sobre las prestaciones de la comunicaciones del entorno monitorizado. Ello es estudiado brevemente en el siguiente apartado.

IV. Análisis de prestaciones

En esta sección se realizará un breve análisis de prestaciones del protocolo propuesto. Para ello, se obtendrá el ancho de banda AB (en bits/s) consumido por la transmisión de los mensajes previamente especificados.

Consideremos una red MANET compuesta de L nodos legítimos $\{N_1, ..., N_L\}$ con un rango de cobertura de r metros, y que se encuentran distribuidos uniformemente en un área de $a \times b$ m^2, con $a, b \gg r$. Asumiendo la existencia de movilidad, cada nodo N_i tendrá su propio conjunto de vecinos V_i. En este escenario general, consideramos adicionalmente la existencia de M nodos maliciosos. Dichos nodos serán excluidos de los cálculos, pues es de suponer que éstos no participarán en actuaciones que tienen como objetivo su detección o aislamiento de la red.

Para el cálculo del ancho de banda consumido será necesario definir una serie de cantidades de interés, así como sus notaciones.

- $\boldsymbol{f_{i,j}^{a/s}}$: representa la frecuencia (en transmisiones por segundo) con la que el nodo N_i envía mensajes (de alerta o de solicitud de información de seguridad) acerca de un nodo N_j. Dicha frecuencia de transmisión vendrá determinada por el procedimiento de detección subyacente implementado.
- $\boldsymbol{P^{a/s/r}}$: representa el tamaño de los paquetes transmitidos (alerta, solicitud o respuesta de información). Dicho tamaño depende de la existencia de datos adicionales en el caso de los mensajes de alerta o del número de parámetros solicitados/respondidos en el caso de los mensajes de intercambio de información.
- $\boldsymbol{E[V_i]}$: denota el número esperado de vecinos del nodo N_i. Dada L/ab la densidad de nodos en el área total, y $(L/ab)\pi r^2$ el número de nodos en el área de cobertura de N_i, es evidente que, restando el propio nodo:

$$E[V_i] = \frac{(L-1)\pi r^2}{ab} \tag{1}$$

- $\boldsymbol{p(I_{v,j})}$: representa la probabilidad de que un nodo N_v dado conozca la información solicitada relativa al nodo N_j y, en consecuencia, pueda responder con un mensaje (*unicast*) a la solicitud recibida.

Una vez definida la notación se ha de distinguir la aplicación concreta para la que se está empleando el protocolo, pues tanto el número como el tamaño de paquetes intercambiados (y con ello el ancho de banda) será dependiente del uso.

IV-A. Notificación de alertas

Para calcular el ancho de banda consumido por la notificación de alertas debemos considerar el peor escenario, es decir, aquel en el que todos los nodos de la red tienen conectividad con al menos otro de los nodos. Puesto que la idea es notificar a todos los nodos la existencia del nodo malicioso, este proceso será *broadcast* a toda la red, donde cada nodo retransmitirá el mensaje de alerta recibido. En esta situación, el número de paquetes de alerta propagados por la red para la notificación iniciada por el nodo N_i relativa al nodo malicioso N_j será de L paquetes (siendo L el número de nodos legítimos en la red).

En consecuencia, el valor esperado del ancho de banda para las situaciones de alerta iniciadas por el nodo N_i respecto a N_j, $AB_{i,j}^{alert}$, será:

$$E[AB_{i,j}^{alert}] = f_{i,j}^{a} \cdot P^{a} \cdot L \text{ bits/s} \quad (2)$$

IV-B. Intercambio de información de seguridad con vecinos

Con respecto a la segunda aplicación aquí prevista, el intercambio de información se producirá cada vez que un nodo N_i precise conseguir información de seguridad acerca de un nodo N_j para, por ejemplo, determinar su comportamiento. Dicho flujo se inicia con un mensaje de solicitud *broadcast* a los vecinos, que será respondido únicamente por aquellos que conozcan la información solicitada por el iniciante. Dichas respuestas serán enviadas en mensajes de respuesta *unicast* (véase Sección III-B).

Así, el valor esperado del ancho de banda consumido ante las posibles peticiones de información de un nodo N_i a sus vecinos respecto del nodo N_j, $AB_{i,j}^{inf}$, será:

$$E[AB_{i,j}^{inf}] = f_{i,j}^{s} \cdot \left(P^{s} + P^{r} \cdot E[V_i] \cdot p(I_{v,j}) \right) \text{ bits/s} \quad (3)$$

Una vez que completemos la implementación efectiva del protocolo propuesto, estos estudios teóricos deberán concretarse sobre escenarios prácticos a fin de ser conscientes de los requisitos reales involucrados.

V. Conclusión

En este trabajo se propone un protocolo de notificación y alerta de eventos de seguridad ideado para la comunicación de actividades maliciosas contra la seguridad de un entorno de red. Por una parte, el procedimiento permite proporcionar diversa información útil, de cara a la adopción de medidas reactivas subsiguientes. Por otro lado, la notificación realizada es distribuida a fin de permitir su uso en entornos no centralizados como son las redes ad-hoc. Por último, el protocolo puede ser usado también como mecanismo de intercambio de información de seguridad entre nodos en este tipo de entornos, con el objeto de posiblitar una detección colaborativa.

Si bien las bondades de la propuesta han sido evidenciadas a nivel teórico en el documento, es objetivo inmediato de los autores la implementación efectiva del protocolo y evaluación de prestaciones del mismo en escenarios experimentales de simulación. Este desarrollo prevé incorporarse al *framework* NETA (*NETwork Attack*) [23], creado por el grupo de investigación NESG (”Network Engineering & Security Group”; http://nesg.ugr.es) y consistente en un entorno basado en OMNET++ para el despliegue, estudio y evaluación de ataques en redes MANET.

Agradecimientos

Este trabajo ha sido parcialmente financiado por el MICINN a través del proyecto TEC2011-22579 y por el MECD a través de la beca del programa de “Formación de Profesorado Universitario” (FPU, Ref.: AP2009-2926).

Referencias

[1] J. He, Mr. Ji, Y. Li, Y. Pan, “Wireless ad-hoc and Sensor Networks: Management, Performance, and Applications,” Boca Raton, FL: CRC Press, 2014.
[2] K.I. Lakhtaria (Ed.), “Technological Advancements and Applications in Mobile Ad-Hoc Networks: Reseach Trends,” Hershey, PA: IGI Global, 2012.
[3] R. Beyah, J. McNair, C. Corbett “Security in Ad-hoc and Sensor Networks,” Hackensack, NJ: World Scientific, 2010.
[4] P. García-Teodoro, L. Sánchez-Casado, G. Maciá-Fernández, “Taxonomy and Holistic Detection of Security Attacks in MANETs,” capítulo del libro “Security for Multihop Wireless Networks”(S. Khan, J. LLoret, Eds.), CRC Press, 2014.
[5] H. Bidgoli (Ed.), “Book of Information Security. Threats, Vulnerabilities, Prevention, Detection, and Management. Volume 3,” John Wiley & Sons, 2006.
[6] A. Nadeem, M.P. Howarth, “A Survey of MANET Intrusion Detection & Prevention Approaches for Network Layer Attacks,” en *IEEE Communications Surveys & Tutorials*, Vol. 15, N. 4, 2013, pp. 2027–2045.
[7] B. Feinstein,G. Matthews, “The Intrusion Detection Exchange Protocol (IDXP),” RFC 4767, 2007.
[8] H. Debar, D. Curry, B. Feinstein, “The Intrusion Detection Message Exchange Format (IDMEF),” RFC 4765, 2007.
[9] K. Gorzelak, T. Grudziecki, P. Jacewicz, P. Jaroszewski, L. Juszczyk, P. Kijewski, “Proactive Detection of Network Security Incidents,” ENISA report (A. Belasovs, Ed.), 2011.
[10] R. Danyliw, J. Meijer, Y. Demchenko, “The Incident Object Description Exchange Format,” RFC 5070, 2007.
[11] X-ARF, “Network Abuse Reporting” http://www.x-arf.org, 2014.
[12] P. Saint-Andre, “Extensible Messaging and Presence Protocol (XMPP): Core,” RFC 6120, 2011.
[13] P. Saint-Andre, “Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence,” RFC 6121, 2011.
[14] P. Saint-Andre, “Extensible Messaging and Presence Protocol (XMPP): Address Format,” RFC 6122, 2011.
[15] R. Gerhards, “The Syslog Protocol,” RFC 5424, 2009.
[16] J. Li, S. Khan, Q. Li “An Efficient Event Delivery Scheme in Mobile Ad-hoc Communities,” en *International Journal of Communication Networks and Distributed Systems*, Vol. 10, N. 1, 2013, pp. 25-39.
[17] C. Perkins,E. Belding-Royer, S. Das, “Ad-hoc On-demand Distance Vector (AODV) Routing,” RFC 3561, 2003.
[18] L. Sánchez-Casado, G. Maciá-Fernández, P. García-Teodoro, “An Efficient Cross-Layer Approach for Malicious Packet Dropping Detection in MANETs,” en *11th. IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications (TrustComm)*, Liverpool (UK), junio 2012, pp. 231-238.
[19] L. Sánchez-Casado, G. Maciá-Fernández, P. García-Teodoro, N. Aschenbruck, “A Novel Collaborative Approach for Sinkhole Detection in MANETs,” en *Workshop on Security on ad-hoc Networks (SecAN)*, Benidorm (España), junio 2014, pp. 1-14.
[20] R. Magán-Carrión, F. Pulido-Pulido, J. Camacho-Páez, P. García-Teodoro, “Tampered Data Recovery in WSNs through Dynamic PCA and Variable Routing Strategies,” en *3rd. Int. Conference on Communications and Network Security (ICCNS)*, Londres (UK), noviembre 2013. Publicado en *Journal of Communications*, Vol. 8, N. 11, 2013, pp. 738-750.
[21] R. Magán-Carrión, J. Camacho-Páez, P. García-Teodoro, “A Multiagent Self-healing System against Security Incidents in MANETs,” en *Workshop on Active Security through Multi-Agent Systems (WASMAS)*, Salamanca (España), junio 2014, pp. 1-12.
[22] P. García, J.E. Díaz-Verdejo, G. Maciá, E. Vázquez, “Anomaly-based Network Intrusion Detection: Techniques, Systems and Challenges,” en *Computers & Security*, Vol. 28, 2009, pp. 18-28.
[23] L. Sánchez-Casado, R.A. Rodríguez-Gómez, R. Magán-Carrión, G. Maciá-Fernández, “NETA: Evaluating the Effects of NETwork Attacks. MANETs as a Case Study,” en *Advances in Security of Information and Communication Networks*, ser. Communications in Computer and Information Science, Vol. 381, (A. Awad, A. Hassanien, K. Baba, Eds.), Springer Berlin Heidelberg, 2013, pp. 1-10.

Implementación de un ataque DoS a redes WPAN 802.15.4

Aleix Dorca
Estudis d'Informàtica,
multimedia i Teleciomunicació
Universitat Oberta de Catalunya
Email: adorca@uoc.edu

Jordi Serra-Ruiz
Estudis d'Informàtica,
multimedia i Teleciomunicació
Universitat Oberta de Catalunya
Email: jserrai@uoc.edu

***Resumen*—Las redes industriales y, concretamente, las redes de sensores son hoy en día una realidad emergente y con muchas expectativas de cara al futuro sobre todo en entornos empresariales o públicos. Los grandes ayuntamientos están creando las *smart cities* con este tipo de estructuras. Entre los estándares que existen parece que hay dos que se imponen, los estándares 802.15.4 y ZigBee. Conjuntamente proporcionan un conjunto de protocolos y servicios a los usuarios para permitir una comunicación fiable y segura. Todo y eso, como la mayoría de redes inalámbricas, estas redes no están exentas de potenciales peligros que pueden ponerlas en un compromiso. El objetivo de este artículo es mostrar la implementación de un ataque de denegación de servicio mediante un caso real. Como prueba de concepto, se muestra como dejar inhabilitado un nodo de una red que utiliza la especificación 802.15.4.**

***Palabras clave*—Seguridad (security), Wi-Fi, 802.15.4, ZigBee, Smart Cities, DOS.**

I. Introducción

Antes de tratar específicamente los estándares que son el objeto de este artículo es interesante hacer una breve presentación de lo que son las redes adhoc y, como caso especial, las redes de sensores que se utilizan es este tipo concreto de infraestructuras. Estos dos tipos de redes tienen muchas similitudes pero a la vez también presentan algunas diferencias importantes.

[1] describe una red adhoc como un conjunto de nodos que se comunican entre sí mediante unos enlaces radioeléctricos. Cada dispositivo de esta red tiene libertad total de movimientos por el espacio, y eso hace que la red se tenga que adaptar a los cambios de manera autónoma y automática. Los nodos pueden aparecer y desaparecer en cualquier momento. Por eso cada nodo se tiene que comportar como un encaminador de la información para el resto de los nodos, ya que los cambios en la estructura de la red pueden necesitar de esa característica, y tiene que hacer circular por la red el tránsito que recibe y del cual no es el destinatario final.

Las redes de sensores son una particularidad formada por dispositivos autónomos que por lo general se encargan de monitorizar condiciones ambientales o físicas. Así, por ejemplo, podemos encontrar sensores destinados en el control de temperatura, presión, peso, sonido, vibraciones, etc. Estos dispositivos inalámbricos envían la información que sus sensores detectan a través de la red hacia nodos de control. Estas redes son interesantes puesto que implementar una solución similar utilizando redes cableadas podría suponer un problema de presupuesto y un problema de logística haciéndola inviable en la mayoría de casos. Utilizando el aire como medio de comunicación da la posibilidad de instalar todos aquellos dispositivos o nodos necesarios a un precio muy económico en comparación con levantar toda una calle para colocar unos sensores de presencia de vehículos estacionados.

A grandes rasgos, los dos tipos de redes presentan las siguientes similitudes:

- Permiten la comunicación entre dispositivos mediante el envío de datos con encaminamiento multi-salto (*multi-hop*).
- Lo más usual es tratar con dispositivos que disponen de recursos mínimos, ya sea de proceso, memoria o almacenamiento, en los dos casos. Los dispositivos acostumbran a ser pequeños y alimentados con pequeñas baterías que con el tiempo hay que reemplazar.

Las principales diferencias son las siguientes:

- Las redes adhoc permiten la comunicación entre cualquier par de dispositivos mientras que las redes de sensores definen tipos de encaminamiento específicos.
- A pesar de que los dispositivos acostumbran a tener recursos mínimos, esta característica se hace todavía más patente en las redes de sensores donde los dispositivos, una vez asociados a la red, han que estar largos períodos tiempos (meses o años) sin ser recargados o reemplazados. Es evidente que la gestión de la energía es un punto clave en la gestión y diseño de estas redes. Algunos ejemplos podrían ser:
  - ◇ Calles donde los sensores de ocupación de plazas de estacionamiento se encuentran bajo tierra.
  - ◇ Monitorización de espacios naturales. (humedad, luz, lluvia...)
  - ◇ Detección de incendios, terremotos o inundaciones.
  - ◇ Control del tráfico.
- Los nodos en redes de sensores a menudo tienen relaciones de confianza entre nodos cercanos puesto que no es extraño que todos ellos recojan información similar o redundante, por lo que enviarla por la red sería una pérdida de recursos. Este comportamiento de complicidad no se

encuentra en las redes adhoc.

I-A. Protocolo 802.15.4 y ZigBee

En esta sección se mostrará el comportamiento general del estándar 802.15.4 y el protocolo ZigBee, que permiten comunicar dispositivos remotos de bajo rendimiento.

I-A1. Estándar 802.15.4: El estándar 802.15.4 define las capas de comunicación física y de acceso al medio de la pila de protocolos. Otras características de este estándar son el hecho que presenta una alta flexibilidad en cuanto a la configuración de red, un bajo coste computacional y a la vez un muy bajo consumo [3].

La capa física además de enviar y recibir paquetes a la red se encarga de toda una serie de tareas como por ejemplo la activación del enlace, la detección de energía o el indicador de baja calidad.

Canales de transmisión

La capa física se puede configurar para transmitir en diferentes canales o bandas de frecuencia dependiendo de las necesidades de cada caso. Se definen los siguientes canales: Banda de 2450 MHz de 16 canales con una velocidad máxima de 250 kbps. Banda de 915 MHz de 10 canales con una velocidad máxima de 40 kbps y banda de 868 MHz de 1 canal con una velocidad máxima de 20 kbps.

Se ha que tener en cuenta que la comunicación en la banda de 2450 MHz trabaja en el misma zona de frecuencia que los dispositivos Wi-Fin 802.11. Es por eso que se recomienda escoger los canales 15, 20, 25 o 26 del estándar 802.15.4 para no provocar interferencias.

Capa de acceso al medio

Esta capa define como se realiza la comunicación a bajo nivel entre dispositivos. Se definen aspectos como la generación de los *beacons*, la duración de la transmisión de estos, el establecimiento de una política de *slots* equitativa, la asociación de nodos y la validación de las tramas [4].

El protocolo de acceso al medio se implementa mediante el algoritmo CSMA-CA.

La capa de control de acceso al medio define dos tipos de dispositivos que se pueden encontrar en una red 802.15.4 [5]:

- Los nodos de función completa (*Full Function Device*–FFD): Estos vienen equipados con una serie completa de funciones en la capa de acceso al medio, cosa que les permite actuar como coordinadores de la red o como dispositivos finales. Cuando estos nodos actúan como coordinadores pueden enviar *beacons*, o señalizaciones para proveer la red de servicios de sincronía, comunicación y procesos de acceso a la misma.
- Los nodos de función reducida (*Reduced Function Device*–RFD): Este tipo de nodos solo pueden actuar como nodos finales y no como coordinadores. Vienen equipados con sensores, actuadores, transductores, interruptores, etc. Y solo pueden interactuar con dispositivos que sean nodos de función completa.

Todas las redes 802.15.4 han de tener como mínimo un dispositivo FFD que actúe como coordinador. Uno de estos dispositivos es elegido coordinador de la PAN (*Personal Area Network*), responsable de las tareas de control de la red y de la seguridad.

Cualquier dispositivo RFD siempre tiene que estar asociado a un FFD para el correcto funcionamiento de la red. En el apartado de topología de la red de ZigBee se ven algunos ejemplo de asociación de nodos.

Formato de trama

La tabla I muestra el esquema de la estructura del formato de una trama MAC.

Donde: MHR es la cabecera, MSDU los datos (o *Payload*) y un final de trama (MFR)

- Campo de control: Este campo de longitud 16 bits contiene toda la información de control del paquete. Eso incluye el tipo de trama (Datos, ACK, etc.), si la seguridad está habilitada, si se necesario un ACK para esta trama. Además se define si los campos de direccionamiento estarán todos presentes y la longitud de estos. Por ejemplo, si el campo Intra-PAN está activo entonces el campo de PAN origen no estará presente. La tabla II muestra la estructura de estos campos.
- Control de secuencia: Este campo se utiliza para verificar el orden de llegada de los paquetes y para evitar ataques de reenvío. Este valor aparecerá en los paquetes ACK conforme el paquete con el código de secuencia especificado ha sido recibido.
- Campos de direccionamiento: Estos cuatro valores no son siempre obligatorios y dependerá del tipo de trama. Existen cuatro campos que corresponden a las PAN de origen y destino y la dirección origen y destino a nivel MAC. Las direcciones MAC pueden tener diferentes medidas según los estándares IEEE: 16 o 64 bits.
- Carga (Payload): En este campo se almacena los datos del paquete, concretamente, estará los datos del protocolo ZigBee, a pesar de que no tendría que ser siempre así puesto que este estándar está preparado para encapsular otros protocolos. La longitud de este campo es variable siempre y cuando no sobrepase la longitud máxima de una trama MAC (127 bytes).
- Código de verificación (FCS): 16 bits que almacenan los datos de verificación de la trama. Se utiliza un algoritmo CRC de 16 bits.

I-A2. La especificación ZigBee: La especificación ZigBee se encarga de definir en detalle las capas superiores de la pila de protocolos. Concretamente se trata de las capas de red y de aplicación. Además, ZigBee también define los siguientes aspectos:

- Tipo de dispositivos descritos en el apartado I-A1
- Topología de la red.
- Procedimiento para acceder o abandonar la red.
- Algoritmos de encaminamiento.

Topología

En en cuanto a la topología de la red se definen tres tipos de distribución de los nodos [5]. Se pueden apreciar en la figura 1

- Estrella: Existe un nodo central que a la vez actúa de coordinador y gestor. El nodo central es un dispositivo

Bytes:2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Control	Secuencia	PAN destino	Destino	PAN origen	Origen	Datos	FCS
		Campos de direccionamiento					
MHR						MSDU	MFR

Tabla I
FORMATO DE LA TRAMA MAC

Bits: 0-2	3	4	5	6	7-9	10-11	12-13	14-15
Tipo de trama	Segur. habilitada	Trama pendiente	ACK necesario	Intra PAN	Reser.	Tipo direc. destino	Reser.o	Tipo direc. origen

Tabla II
FORMATO DEL CAMPO DE CONTROL DE LA CAPA MAC

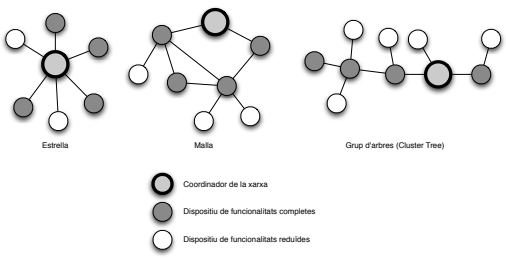


Figura 1. Topología de las redes de sensores

FFD mientras que el resto son, o pueden ser, RFD.

- Malla: Este tipo de topología permite que entre encaminadores FFD del tipo Cluster Tree también haya comunicación sin tener que depender del nodo central.
- Grupo de árboles: En este tipo de topología existe un nodo central que a la vez actúa como coordinador de la red y gestor. De este nodo dependen toda otra serie de nodos que pueden ser tanto FFD como RFD. En este caso los nodos FFD actuarán como encaminadores para otros dispositivos RFD. Es escalable siempre y cuando los encaminadores sean FFD.

Encaminamiento

El encaminamiento en las redes ZigBee se realiza dependiendo del tipo de topología que se ha escogido. En cualquier caso intervienen procesos de descubrimiento de rutas así como encaminamiento mediante tablas de rutas. El algoritmo de encaminamiento es sencillo y se puede resumir en que si la información es para el propio nodo se pasa a la pila de protocolos, sino se mira si es para un nodo hijo y se envía al nodo o a la ruta preestablecida.

Para realizar el descubrimiento de la ruta idónea se utiliza el algoritmo AODV (*AdHoc on Demand Distance Vector Routing*) que consiste en enviar un paquete a todos los nodos vecinos, que también propagarán, con el fin de llegar al nodo destino. A medida que el paquete pasa por los nodos se actualiza el coste de la ruta por la cual el paquete ha pasado. Y cuando el paquete llega al destino se envía una respuesta al nodo origen con la ruta más óptima.

II. ASPECTOS DE SEGURIDAD

Aparte de estas características básicas se ha puesto especial énfasis en la seguridad de los estándares. Las dos especificaciones establecen una serie de opciones que hacen que la seguridad ya no dependa del propio código de la aplicación. El protocolo puede cifrar, por ejemplo, el contenido de las tramas.

II-A. Seguridad en IEEE 802.15.4

El estándar 802.15.4 define tres modos de seguridad [5]–[6]:

1. Sin seguridad.
2. Modo ACL (*Access Control Lists*). No presenta cifrado pero solo se aceptan paquetes de dispositivos en listas de control de acceso.
3. Modo Seguro. Algunas de las características que puede incluir este modo son: Integridad, Confidencialidad, Control de acceso, etc.

A la vez se definen cuatro servicios de seguridad:

1. Control de acceso vía ACL.
2. Cifrado de datos mediante el algoritmo AES de 128bits.
3. Integridad de las tramas.
4. Control de secuencia para evitar ataques de reenvío.

Finalmente se definen ocho configuraciones de seguridad posibles en las que se puede escoger el algoritmo de cifrado así como el modo y la longitud del código de integridad. Algunas configuraciones incluyen solo autenticación de las tramas mientras que otras más completas añaden la opción de cifrado según los requerimientos de la aplicación.

II-B. Seguridad en ZigBee

Aparte de los elementos de seguridad del estándar 802.15.4 de los que ZigBee también se puede beneficiar, define toda una serie de conceptos orientados a la seguridad. De entrada la seguridad en ZigBee se basa en los siguientes principios [7]:

- Simplicidad: Cada capa se encarga de su seguridad.
- Es directa: Las claves de cifrado se intercambian directamente entre origen y destino.
- Extremo a extremo: Los datos circulan cifrados de origen a destino sin tener que ser descifrados en cada salto.

ZigBee define tres tipos diferentes de claves de cifrado que se utilizan, cada una de ellas, en casos muy diferenciados [5]. Los tres tipos de claves son:

1. Clave maestra: esta clave no se utiliza para cifrar información, sino para la generación de otras claves. Esta clave se establece en el momento de la construcción, pero puede ser entrada por el propio usuario o bien asignada por un centro de confianza. Todos los dispositivos disponen de una clave maestra propia y única.
2. Clave de red: la disponen todos los dispositivos de la red y se utiliza para enviar mensajes a toda la red. Los mensajes cifrados *broadcast* se cifran y se descifran mediante esta clave cuando la seguridad está habilitada. Esta clave se establece en el momento de la unión a la red o bien mediante procesos de renovación de claves.
3. Clave de enlace: se utiliza para establecer comunicaciones seguras entre dos dispositivos. Se obtiene a partir de la clave maestra mediante un proceso llamado SKKE.

Los servicios de seguridad en ZigBee incluyen métodos para el establecimiento de claves, el envío de estas claves, la protección de tramas y la gestión de dispositivos. Como el control de secuencia (*freshness*), que mediante contadores que se regeneran cada vez que se renuevan las claves se permite controlar la secuencia de los mensajes para que no se realicen ataques de reenvío. La integridad de los mensajes, que asegura que los mensajes enviados no han sido modificados durante la transmisión por ningún tercero. La autenticación, que mediante claves de red o enlace, los dispositivos pueden estar seguros que el origen de los mensajes es de quién dicen ser, evitando la suplantación por parte de intrusos. El cifrado, que mediante el algoritmo AES de 128 bits la protección se extiende a nivel de red o dispositivo. El cifrado es opcional sin necesidad de afectar otras características de seguridad. Las tramas están encapsuladas en la especificación 802.15.4, por lo que las cabeceras no van cifradas, como se puede observar en la tabla I

II-C. Vulnerabilidades

Las redes adhoc y, por extensión, las redes de sensores pueden ser vulnerables a toda una serie de ataques que se pueden categorizar de la siguiente manera [2]–[5]:

1. Denegación de servicio (*Denial of Service*-DOS): Estos ataques hacen que un nodo deje de funcionar mientras dura el ataque o indefinidamente.
2. Escucha de la red (*eavesdropping*): Como su nombre indica, un dispositivo escucha la red a la espera de recibir información. Evidentemente utilizando cifrado en la red o sobre los datos este ataque pasa a ser inútil, siempre y cuando no se combine con algún ataque para obtener las claves de cifrado.
3. Usurpación de identidad (*spoofinng*): Este ataque consiste al hacerse pasar por otro dispositivo, ya sea a nivel MAC, de red u otras. De este modo se pueden obtener paquetes por el dispositivo atacante. Si este usurpa un encaminador y actúa de manera ”legal”, podrá capturar toda la información que encamine.
4. Reenvío de paquetes (*replay*): Este ataque consiste a reenviar paquetes capturados para que el destino actúe de manera errónea. Por ejemplo, si un sensor manda un mensaje de incremento de temperatura, el nodo atacante podría reenviar un decremento del valor provocando que el nodo destino actúe de manera inversa a la deseada. Para evitar este tipo de ataques se utilizan los códigos de secuencia, el cifrado, etc.

III. Ataques

En esta sección se muestran la descripción de los ataques que se han realizado en este trabajo.

III-A. Ataques de denegación de servicio

Los ataques de denegación de servicio se pueden categorizar según la capa de la pila de protocolos a la que van dirigidos [8].

Posibles ataques a la capa física:

- Interferencias (Jamming): consiste en saturar un canal de comunicación con información errónea para que ningún otro dispositivo pueda utilizarlo. En general este tipo de ataque se cancela mediante diferentes canales en los que transmitir.
- Alteración de datos (*Data tampering*): consiste en modificar la información que circula por la red, capturando los datos y modificándolos. Este tipo de ataque se puede frenar con códigos de verificación de datos.

Posibles ataques a la capa de enlace:

- Colisión: en este caso, similar al *jamming*, se modifica cierta información del origen provocando que la verificación del paquete provoque un error. De este modo se provoca un reenvío de los paquetes que puede llevar al límite los recursos. No se conoce un procedimiento totalmente fiable para evitar este tipo de ataques.
- Ruido en el canal: existen muchos errores en la transmisión que implican un gran reenvío de paquetes. Si se consigue que un dispositivo agote todos sus recursos y quede aislado o inoperativo el ataque se considera satisfactorio. Para evitarlo se puede establecer un umbral a partir del cual no se retransmite.
- Longitud de las tramas: este ataque deja la red inservible ocupando el canal enviando muy poca información a intervalos regulares y constantes.

Posibles ataques a la capa de red y encaminamiento:

- *Homing*: este ataque obtiene información sobre nodos que son de especial importancia en la red. En las redes ZigBee, el ataque se centra en el PAN Coordinator. El cifrado de datos puede mitigar el ataque.
- Encaminamientos erróneos o selectivos: se implementa sobre encaminadores que rechazan o encaminan erróneamente paquetes. Para evitar este tipo de ataques el dispositivo puede probar de encaminar los paquetes por otra ruta.
- Agujeros negros (*Black holes*) y agujeros de gusano (*Wormholes*): en las redes que utilizan el protocolo de descubrimiento de rutas basado en el coste del enlace, este ataque puede provocar la construcción de rutas erróneas si un dispositivo siempre anuncia la calidad de su enlace como la mejor. De este modo la mayoría de

Figura 2. mota Z1 de Zolertia

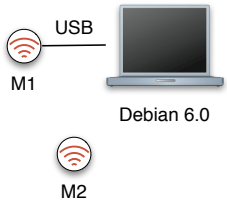


Figura 3. Esquema de montaje

paquetes serán enviados a través de él y este podrá aplicar decisiones de encaminamiento erróneas o simplemente descartarlos.

- *Sybil*: se basa en que un nodo pueda presentar varias identidades a la vez. Así se pueden romper esquemas de encaminamiento múltiple o bien causar problemas en entornos geográficamente dispares.

Posibles ataques a la capa de aplicación:

- Inundar la red (*Flooding*), *HELLO attacks*: una vez introducido el nodo malicioso en la red, envía peticiones de conexión que pueden llevar al nodo remoto a agotar los recursos y quedar inoperantivo. Para evitar este ataque se presentan soluciones del tipo limitar el número de conexiones establecidas o presentar rompecabezas al cliente que tiene que resolver antes no se le otorgue el recurso.
- De-sincronización: Mediante el envío con códigos de secuencia erróneos a nodos que ha establecido conexión con un tercero se puede forzar el reenvío de tramas. Si además se sigue el ataque con insistencia se pueden agotar sus recursos. Estos ataques no tienen sentido si los nodos pueden comprobar la veracidad de los paquetes mediante cifrado o códigos MAC.

IV. Resultados experimentales

Este apartado pretende mostrar una prueba de concepto en el que se ha llevado a cabo un ataque de denegación de servicio sobre una red 802.15.4/ZigBee. El proceso que se ha seguido es muy simple y lo que se desea mostrar es la facilidad con la que ha sido posible dejar sin recursos un nodo de la red.

Entre los posibles ataques que se han mostrado, este ataque recaería sobre el agotamiento de recursos en la capa de aplicación y la capa de enlace puesto que ambos tienen gran parte de implicación.

Para montar la red ZigBee se han utilizado dos sensores o *motas*. Concretamente se ha utilizado la plataforma Z1 de la marca Zolertia [9], mostrada en la figura 2.

Por otro lado, para simular el dispositivo atacante se ha utilizado un sniffer/inyector de la marca Atmel: el dispositivo RZUSBSTICK [10].

En esta prueba de concepto es necesario que un sensor actúe como nodo encaminador (llamado M1 en la figura 3) y que el otro actúe como dispositivo RFD hoja (M2). La topología escogida es la de estrella. Y el nodo al que se atacará agotando los recursos es el que actúa como encaminador (la mota M1). Por lo que en el caso de tener más sensores conectados a la mota M1, todos dejarían de poderse comunicar. Para programar las motas se han de conectar inicialmente a la máquina Debian con un cable USB/MicroUSB, después ya no es necesario esa conexión y pueden operar mediante pilas y por tanto ser un nodo completamente autónomo. Con las aplicaciones base que hay disponibles, la mota M1 se ha programado con la aplicación IPBaseStation, mientras que la mota M2 utiliza el servicio TCPEcho.

El siguiente paso consiste en examinar los paquetes que se envían en el momento de la asociación de la mota M2 en la red y los que se envían durante la transmisión de paquetes al servicio Echo para poder, posteriormente, inyectar los paquetes previamente modificados desde el dispositivo atacante.

Para averiguar qué paquetes circulan por la red durante las fases de asociación de dispositivos y la conexión al servicio Echo se ha utilizado el software Wireshark y la herramienta zbdump. Y para escuchar la red mediante el dispositivo RZUSBSTICK son necesarias las herramientas que se encuentran en KillerBee [11]. Eso es debido a que una simple tarjeta WiFi convencional no puede escuchar las frecuencias de las redes 802.15.4

Mediante el dispositivo RZUSBSTICK, el software Wireshark y la herramienta zbdump se ha determinado cuáles son los paquetes esenciales y necesarios para llevar a cabo la asociación de un dispositivo al encaminador. Y al mismo tiempo se ha obtenido el paquete que se envía cuando se solicita un Echo al servicio de la mota M2. Esta información se ha utilizado para construir posteriormente unos paquetes específicos con los que atacar el sistema.

IV-A. Atacando al sistema

El ataque desarrollado se basa en dos scripts que por un lado asocian una serie de nodos falsos al encaminador M1 y de la otra envían paquetes al servicio TCPEcho de la mota M2 de manera ininterrumpida mediante el encaminador M1. Los paquetes de la Mota M2 con las respuestas solicitadas pasan a través de M1 y llegan a los nodos falsos maliciosos que están programados especialmente para no responder a las peticiones ACK que solicita M2 de sus respuestas. Como el número de nuevas peticiones por parte de los nodos falsos no cesa y la mota M2 está continuamente enviando los datos que supone no han llegado a los nodos destino, la tabla del encaminador se satura y se queda sin recursos, lo que hace bloquear completamente el encaminador, dejando a todos los

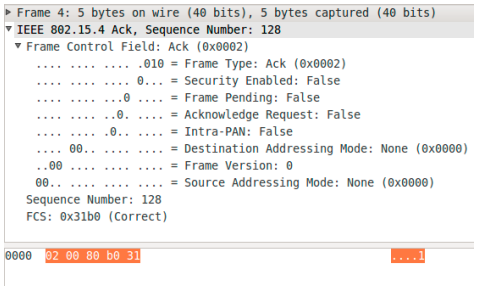


Figura 4. Captura de red, ACK

sensores aislados de la red.

La programación del capturador de tráfico es relativamente sencilla, ya que se aprovecha las tramas reales para inyectar el código modificado y no contestar a los ACK. Básicamente se realizan los siguientes pasos:

1. Inicialización: utilizando la API de Killerbee se define en qué canal retransmitir y el período de tiempo que se dejará pasar entre paquetes. Las herramientas KillerBee todavía no disponen de un programa que escuche todos los canales a la vez por el que se ha de determinar. Para este escenario se ha fijado el canal.
2. Bucle principal: para cada dispositivo se genera el valor del paquete en la variable *origen*. La posición donde se ha de poner este valor se ha determinado del estudio de los paquetes capturados con zbdump y Wireshark y de la especificación del formato de trama MAC. En la figura 4 se puede ver un ejemplo de los paquetes capturados, concretamente, el Ack correspondiente en formato Daintree mediante la herramienta zbdump.
3. Inyección: el paquete se envía a la red(inject).
4. Se duerme el sensor los segundos especificados.
5. El script acaba con la limpieza de la trama enviada.

En el caso del ataque con el envío de solicitudes el script que hace las peticiones al servicio Echo es similar al anterior caso, cambiando únicamente las variables utilizadas de la API.

Una vez ejecutado el ataque el nodo falso envía paquetes de solicitud de Echo de manera indiscriminada desde direcciones falsas de dispositivo asociados a la M1 que no existen provocando que la M1 tenga que responder a todos los Ack solicitados, así como a la propia respuesta del protocolo. Como el nodo falso malicioso no responde a ningún paquete, se crea una situación en la que la mota M1 tiene que reenviar paquetes varias veces.

Cuando se desea realizar una comunicación con la mota M2 esta responde correctamente a algunas peticiones pero a medida que el ataque progresa ésta dejará de responder. Pero en realidad la mota M1 ha dejado de encaminar paquetes. El número de paquetes necesario para que la mota M1 deje de responder varía en cada ejecución y depende de la cantidad de envíos correctos realizados y del tiempo transcurrido antes del ataque.

En general, según las pruebas llevadas a cabo, la mota M2 responde una decena de peticiones de Echo antes de que la mota M1 deje de responder. El bloqueo de la mota M1 es absoluto siendo necesario un reinicio completo del dispositivo para que vuelva a funcionar la comunicación con la mota M2.

El envío de paquetes por parte del nodo malicioso no es en ningún caso exhaustivo. En este caso cada paquete se envía con un retraso de 0.5 segundos respeto la anterior cosa que permite un tiempo suficiente a todas las partes del sistema a responder sin provocar un bloqueo del medio. A pesar de que el ataque de saturación del medio es igualmente factible, este no es demasiado interesante para los protocolos.

V. Conclusión

En este artículo se ha descrito el funcionamiento y las características básicas de los estándares 802.15.4 y ZigBee, el montaje y puesta en funcionamiento de redes de sensores utilizados en redes de ”Smart Cities”.

A continuación se han descrito las opciones de seguridad que estos dos protocolos ofrecen, describiendo cuales son los casos de ataque más comunes en este tipo de redes y dedicando un apartado especial a los ataques de denegación de servicio en el que se basa la prueba de concepto de ataque sobre una red 802.15.4/ZigBee.

Finalmente se ha ejemplificado un caso real en el que es posible dejar sin recursos un dispositivo 802.15.4 mediante únicamente herramientas de libre distribución y la programación en Python de un script que envía solicitudes Echo de manera indefinida hasta que el dispositivo se satura y deja de funcionar.

Como trabajo futuro está el estudio más detallado de las posibles mejoras al protocolo estándar para que no se puedan realizar este tipo de ataques sobre las cabeceras no cifradas.

Agradecimientos

This work was partly funded by the Spanish Government through projects TIN2011-27076-C03-02 “CO-PRIVACY” and CONSOLIDER INGENIO 2010 CSD2007-0004 “ARES”.

Referencias

[1] E. Peig, “Redes abiertas (Cuando los usuarios forman parte de la red),” Barcelona, Ed. UOC, 2012.
[2] C. Karlof, D. Wagner, “Secure routing in wireless sensor networks: Attacks and countermeasures,” en *IEEE SPNA*, 2002.
[3] S. Coleri Ergen, “ZigBee/IEEE 802.15.4 summary,” Unknown, 2004.
[4] IEEE, “802.15.4,” http://www.ieee802.org/15/pub/TG4.html.
[5] P. Baronti, P. Pillai, V.W.C. Chook, S. Chessa, A. Gotta, and Y. Fun Hu, “Wireless sensor networks: A survey on the state of the art and the 802.15.4 and ZigBee standards,” *Computer Comunications*, vol 30(7), pp. 1655–1695, 2007.
[6] H. Li, B. Xue, W. Song, “Application and Analysis of IEEE 802.14.5 Security Services.” en *2nd International Conference on Networking and Digital Society (ICNDS)*, pp. 139–142, 2010.
[7] H. Li, Z. Jia, X. Xue, “Application and Analysis of Zigbee Security Services Specication,” en *Networks Security Wireless Communications and Trusted Computing (NSWCTC)*, pp.494–497, 2010.
[8] A.D. Wood, J.A. Stankovic, “Denial of service in sensor networks”, *Computer*, vol.35(10), pp.54–62, 2002.
[9] Zolertia, “Platform Z1”, http://www.zolertia.com/ti
[10] Atmel, “Rzusbstick”, http://www.atmel.com/tools/rzusbstick.aspx
[11] KillerBee, “Homepage”, http://code.google.com/p/killerbee/

Análisis y Desarrollo de un Canal Encubierto en una Red de Sensores

Jose A. Onieva
Dpto. de Lenguajes y Ciencias de la Computación
Universidad de Málaga
Email: onieva@lcc.uma.es

Ruben Rios
Dpto. de Lenguajes y Ciencias de la Computación
Universidad de Málaga
Email: ruben@lcc.uma.es

Bernardo Palenciano
Dpto. de Infraestructura de TTI
Algeciras
Email: bpalenciano@ttialgeciras.com

***Resumen*—Continuamente aparecen nuevos estudios así como nuevos desarrollos de canales encubiertos. Como veremos, existen más de cien diseños distintos para redes de ordenadores, pero no hemos encontrado en la literatura ningún análisis, diseño e implementación de canales encubiertos sobre redes de sensores. En este artículo presentamos los resultados del diseño e implementación de un canal multitasa basado en los tiempos de monitorización sobre una red de sensores. En este proceso se han establecido las principales propiedades necesarias y, en base a ellas, se desarrolla e implementa el canal encubierto. Se describe el proceso de desarrollo y se analiza su detectabilidad.**

***Palabras clave*—Canales encubiertos (covert channels), Detección de intrusos (Intrusion detection), Information Warfare, Redes de sensores (sensor networks), Seguridad de la Información (Information Security), Seguridad en redes (Network Security).**

I. Introducción

La noción de canal encubierto surgió hace varias décadas en el contexto de los sistemas de seguridad multinivel [1], donde procesos con distintos niveles de seguridad no deberían comunicarse entre sí. De esta forma, los canales encubiertos pueden definirse como "cualquier canal de comunicación que puede ser aprovechado por un proceso para transferir información de manera que viola la política de seguridad del sistema" [2]. Siendo una propiedad fundamental de estos canales que su presencia pase inadvertida ante un posible observador.

Si bien los canales encubiertos nacen en el contexto de los sistemas de seguridad multinivel, el ámbito de estudio fue evolucionando a medida que los sistemas se conectaban entre sí, dando origen a canales encubiertos en redes de comunicación [3]. No obstante, hasta donde alcanza nuestro conocimiento, no existen diseños en *redes de sensores*.

Quizás el estudio más próximo a nuestro trabajo se encuentre en [4], donde los autores presentan un análisis y diseño de canales encubiertos en protocolos de enrutamiento dinámico para redes ad-hoc. En efecto, las redes de sensores pueden ser consideradas como un tipo de red ad-hoc, pero presentan además un gran número de características específicas que obligan a focalizar el análisis y diseño de este tipo de canales sobre ellas. Más aún cuando las redes de sensores están siendo cada vez más utilizadas para la monitorización y control de individuos, ambientes y recursos en multitud de escenarios, tanto militares como civiles.

En este trabajo comenzamos ofreciendo una visión general de las características y particularidades tanto de los canales encubiertos como de las redes de sensores (sección II). Seguidamente, en la sección III analizamos los requisitos necesarios que debería ofrecer un canal encubierto basado en redes de sensores a partir de un escenario ficticio. Asimismo, presentamos el diseño de un canal encubierto multitasa que se ajusta a los requisitos establecidos anteriormente. En la sección IV demostramos la viabilidad del diseño a partir de una prueba de concepto sobre una red de sensores. A continuación se presenta un analisis de la detectabilidad del canal (sección V). Por último, se presentan las conclusiones y posibles líneas de trabajo futuro.

II. Preliminares

II-A. Canales Encubiertos

Los canales encubiertos pertenecen al campo de la ocultación de la información. A diferencia de la criptografía, que se preocupa de mantener desconocido el significado de la información, esta disciplina tiene como objetivo evitar el descubrimiento de la información en sí.

Para entender mejor el concepto de canal encubierto se suele acudir al problema de los prisioneros (*prisioners' problem* [5]): Alice y Bob se encuentran en prisión y están intentando desarrollar un plan para escapar. Se les permite comunicarse a condición de que Walter, el guardián, pueda inspeccionar todas las notas que se intercambian. En el caso de que Walter detectara algún indicio de que Alice y Bob están planeando fugarse, éste no les permitiría seguir comunicándose. Así pues, podemos ver que los canales encubiertos facilitan un medio de comunicación que pase inadvertido a los ojos de un posible examinador del contenido o del formato en el que se realiza una comunicación aparentemente normal.

Existen numerosos estudios que aglutinan y clasifican los distintos tipos de canales en base a diversos criterios [6], [7]. Sin embargo, la clasificación más utilizada, y que adoptamos en este artículo, se fundamenta en las técnicas de ocultación utilizadas:

- Canales de almacenamiento (*Storage Channels*): son aquellos que permiten que un proceso escriba en una zona de memoria para que otro proceso recupere tal información mediante la lectura de tal zona.

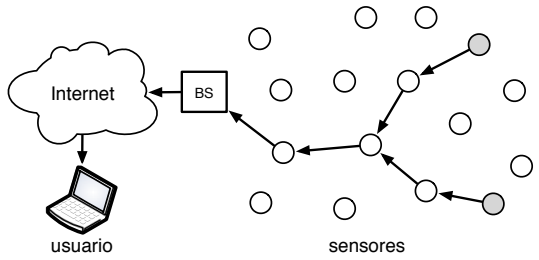


Figura 1. Red de sensores

- Canales de temporización (*Timing Channels*): en estos canales un proceso codifica información mediante la modulación de su propio comportamiento de manera que otro proceso al observar estos cambios es capaz de recuperar la información señalizada.

Si bien esta clasificación es muy general y está orientada a procesos, es igualmente válida para canales encubiertos en redes de comunicación, teniendo en cuenta que las zonas de memoria referidas en los canales de almacenamiento se corresponden a determinados campos de los paquetes de red y la modulación del comportamiento en los canales de temporización puede hacerse evidente cambiando la tasa de envío de paquetes.

II-B. Redes de Sensores

Una red de sensores [8] es un sistema distribuido formado por un gran número de dispositivos de capacidad y tamaño reducido (denominados *nodos* o *motas*) cuyo objetivo es monitorizar un determinado fenómeno físico gracias a los sensores que incorporan. Una vez detectada información de relevancia, ésta es transmitida de manera inalámbrica hasta un dispositivo (conocido como estación base o *sink*) que se encarga de procesarla y ofrecerla a los usuarios de la red, como se muestra en la Figura 1.

Dado que la estación base es la encargada de obtener toda la información de los sensores, el modelo de comunicación más habitual es de muchos a uno, donde caben dos opciones dependiendo del número de sensores desplegados y el rango de transmisión de estos. En el modelo de un *único salto* todos los nodos de la red transmiten directamente a la estación base sin necesidad de realizar enrutamiento. Es más simple, pero sólo es posible cuando el número de nodos y el área de despliegue es reducida. En el modelo *multisalto*, los nodos que se encuentran alejados utilizan a sus vecinos para hacer llegar sus datos a la estación base. Normalmente utilizando el camino más corto o protocolos de inundación.

Por otro lado, existen diferentes modos de funcionamiento o de notificación de eventos. Por lo general, suelen distinguirse las tres alternativas [9]. En la monitorización *continua* los nodos envían información sobre el entorno de manera periódica, mientras que en el modelo de monitorización *basado en eventos* se transmite información únicamente cuando un parámetro alcanza un valor excepcional (i.e., ocurre un evento). Además, existe un modo de monitorización *basado en consultas* en el que los nodos responden a las consultas realizadas por los usuarios de la red.

La versatilidad de estas redes y su reducido tamaño hace de ellas una solución ideal para multitud de aplicaciones de monitorización y control, donde los dispositivos se integran discretamente en el entorno. De hecho, este tipo de redes ya ha sido aplicada con éxito en multitud de escenarios [10], lo que las convierte en sistemas cada vez más aceptados.

Al mismo tiempo, esto puede suponer un mayor interés por explotar este tipo de redes con fines maliciosos. Se hace necesario por tanto un minucioso estudio sobre la viabilidad de desarrollar canales encubiertos en redes de sensores, pues su utilización repercutiría negativamente en la seguridad y privacidad de los entornos donde se desplieguen estos sistemas.

III. Estudio y Análisis

III-A. Escenario

Para la creación de un canal de comunicación oculto, en primer lugar, es necesario plantearse su aplicabilidad, y para ello lo más indicado es idear un escenario de uso ficticio en el que podría ser utilizado.

Supongamos una empresa dedicada al cultivo y venta de mejillones en el Estrecho de Gibraltar. La empresa hace uso de una red de sensores en la zona con el objeto de monitorizar las condiciones del medio marino y conseguir así un mejor producto. Sin embargo, ésta no es la única actividad desarrollada por la compañía. Aprovecha su situación privilegiada para llevar a cabo un transporte ilegal de sustancias en contenedores.

Supongamos además que Alice y Bob son agentes de la Guardia Civil que sospechan de las actividades ilegales desarrolladas por la empresa citada anteriormente. Con el fin de destapar el tráfico de sustancias, Alice se infiltra en la compañía y necesita informar a Bob del contenedor donde se transportan las sustancias ilegales para que pueda atraparlos en el acto. Alice, que teme por su integridad física si fuese descubierta informando de esta actividad, decide idear un mecanismo de comunicación oculta utilizando la red de sensores. Sin embargo, por cuestiones geográficas, Bob sólo tiene acceso a un número limitado de sensores.

Nuestro canal oculto de comunicación debería ser de utilidad para que Alice pueda informar a Bob del contenedor en el que se encuentran las sustancias ilegales sin ser delatada.

III-B. Requisitos del canal

A continuación analizamos los requisitos que serían deseables para el tipo de canal encubierto que necesitarían Alice y Bob a fin de alcanzar su objetivo en el escenario propuesto.

- Grado de detectabilidad. El canal de comunicación oculto debe ser difícilmente detectable. Esta característica se ve facilitada por el hecho de ser las redes de sensores un campo bastante inexplorado en la búsqueda de este tipo de canales[1].

[1]Como hemos indicado previamente, en este trabajo presentamos el primer canal encubierto para redes de sensores.

- Ancho de banda moderado. La capacidad del canal no se considera un factor esencial ya que la intención es el envío de pequeñas cantidades de información, como por ejemplo, la referencia de un contenedor.
- Integridad. Debido a que la información enviada es sensible, es necesario que ésta sea recibida correctamente.
- Sentido de la comunicación. Basta con crear un canal unidireccional, pues el propósito no es realizar un intercambio de datos sino, simplemente, comunicar información desde un punto a otro.

Estas particularidades determinarán el ámbito de aplicabilidad del canal, y a su vez éste puede condicionar la decisión del tipo de canal a implementar.

III-C. Configuración de la red

El escenario de aplicación determina el modo de funcionamiento y configuración de la red, y esto, a su vez, influye en el tipo de canal encubierto que es más conveniente desarrollar dado los requisitos establecidos anteriormente.

Debido a que el objetivo de nuestra red de sensores es la de tomar valores de diferentes parámetros del agua cada cierto intervalo de tiempo, el modo de funcionamiento más conveniente es el de monitorización continua.

Por otra parte, dado que los sensores se encontrarán desplegados en un área extensa se hace imposible el uso de un modelo en un único salto, ya que el rango de transmisión de estos haría imposible que se comunicaran directamente con la estación base. Así, el modelo de enrutamiento multisalto es el más adecuado para nuestro escenario.

En cuanto al sistema operativo de la red de sensores, cabe destacar Contiki [11] y TinyOS [12]. En este trabajo nos hemos decantado por Contiki debido a las bondades de su simulador.

III-D. Diseño de un canal multitasa

Debido a que el cambio en la frecuencia de monitorización de los sensores es normal para atender a las distintas circunstancias que se producen en el medio, a los requisitos de consumo de energía y a las necesidades de procesamiento de la estación base, utilizar estos cambios para la implementación de un canal encubierto parece prometedor.

Al diseñar un canal de temporización podemos optar por una canal *binario*, tal y como se hiciera en [13], o podemos inclinarnos por un canal *multitasa*. En un canal binario, cada cierto intervalo de tiempo, el emisor puede enviar un paquete o mantenerse en silencio. El receptor monitoriza cada intervalo de tiempo para determinar si un paquete fue recibido o no. El resultado es un código binario donde un 1 representa la detección de un paquete en el intervalo y un 0 representa la ausencia del mismo. En un canal multitasa, emisor y receptor acuerdan dos conjuntos (intervalos de tiempo, carácter) y la correspondencia entre ellos. Así, cada intervalo de tiempo distinto corresponderá a un único carácter (ver Figura 2). Pueden producirse errores de decodificación si los tiempos de los distintos intervalos son muy parecidos. Es decir, en este último caso haría falta una sincronización muy fiable, o la integridad exigida al canal sería difícil de conseguir.

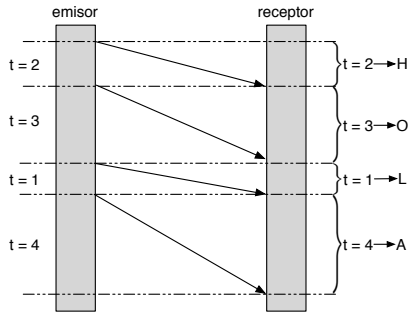
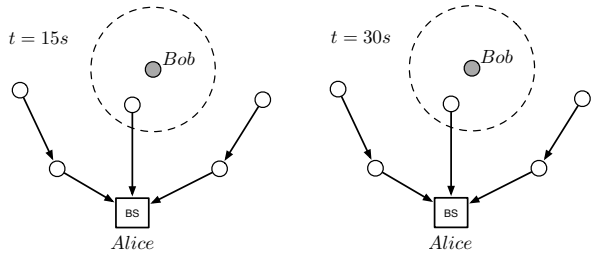


Figura 2. Codificación y sincronización



(a) Estado inicial (canal sin actividad) (b) Cambia el intervalo (canal activo)

Figura 3. Funcionamiento del canal multitasa

Para desarrollar un canal multitasa en nuestra red de sensores, puesto que su funcionamiento es en modo monitorización continua y se envían paquetes de forma periódica siguiendo un tiempo preestablecido t, se tiene que cambiar el tiempo de monitorización según sea necesario. En nuestro escenario, Bob observa constantemente los mensaje enviados en la red de sensores. Mientras la estación base no envíe información oculta, los nodos estarán configurados para enviar mensajes cada intervalo t. En otro caso, la estación base comunicará a los nodos la supuesta intención de reconfigurar los tiempos de monitorización con la excusa de cambiar la estrategia de toma de datos. Los nodos no volverán a comunicarse con la estación base hasta pasado el tiempo ordenado, como se muestra en la Figura 3, donde inicialmente los nodos transmiten cada $15s$ y más tarde se cambia el intervalo de notificación a $30s$. De esta forma, Bob podrá interpretar los distintos cambios en el intervalo de monitorización y finalmente obtendrá su mensaje.

A pesar de que hemos investigado otras posibilidades de ocultación, teniendo también en cuenta la posibilidad de desarrollar canales de almacenamiento, estos últimos suponen cambios que podrían degradar el uso de la red de sensores (e.g. por cambios en los parámetros de enrutamiento) y la vida útil de la misma, además de ser susceptibles a varias técnicas de detección. Si bien para los canales de temporización desarrollados hasta ahora en redes de datos convencionales también se han desarrollado técnicas de detección basadas en la regularidad del envío de los paquetes de datos [14], éstas no son aplicables a las redes de sensores que, por diseño, en el caso de redes de monitorización continua, ya

Tabla I
CODIFICACIÓN DE CARACTERES

Carácter	Tiempo	···	Carácter	Tiempo
FINAL	15	···	f	130
cambio	20	···	z	135
e	25	···	j	140
a	30	···	x	145
(espacio)	35	···	w	150
o	40	···	k	155
s	45	···	···	···

presentan patrones de envío regulares. En cualquier caso, la detectabilidad del canal diseñado se tratará más detenidamente en la sección V.

IV. DESARROLLO DE UN CANAL MULTITASA

El desarrollo de un canal encubierto multitasa requiere establecer una tabla de equivalencias entre intervalos y caracteres. Asimismo, es necesario calcular previamente un intervalo de funcionamiento normal de la red. Este tiempo vendrá determinado por el tipo de aplicación y en nuestro caso lo fijaremos en $t = 15$ segundos, en un intento de alcanzar un balance entre la actualidad de los datos y el tiempo de vida de las motas[2].

Tras realizar pruebas de precisión con el simulador de Contiki (más tarde confirmadas sobre la implementación que hemos realizado con motas Tmote Sky de Moteiv) un nodo que haga las funciones de *sniffer*[3] para Bob tiene un error de precisión de $1 \sim 2$ segs en el cálculo del intervalo de envío de paquetes. Por ello, y para asegurar el requisito de integridad (en contra del ancho de banda) hemos seleccionado un incremento de 5 segundos en los intervalos que diferencian a los distintos caracteres, como puede apreciarse en la tabla I.

En dicha tabla se ha utilizado un conjunto de caracteres reducido del castellano y se ha aplicado una codificación de Huffman [16] según la frecuencia de aparición de estos con el fin de reducir el retraso de las comunicaciones, aumentando así el ancho de banda efectivo del canal. Además, como puede verse, cuando el intervalo de envío de datos por parte de las motas vuelve al original (i.e. 15 segundos), se produce el final del mensaje enviado.

Para implementar el sniffer en el simulador de Contiki es necesario que los nodos de la red se comuniquen en modo broadcast y que sea a nivel de aplicación donde se decida si un paquete está dirigido a un nodo u otro. Es decir, el identificador del nodo destino va incluido en la carga útil de datos o payload de los paquetes pero sólo el autentico destinatario lo procesa cuando observa su identificador. Así, el nodo que hace las funciones de sniffer puede observar y procesar todos los paquetes, aunque no estén dirigidos hacia él a nivel de aplicación.

En nuestro diseño final decidimos utilizar repeticiones con el fin de aumentar la integridad del canal. Por ejemplo, sabiéndose que el intervalo de transmisión de la letra 'a' es de 30 segundos, podemos estar seguros de que pasados 90 segundos desde que la estación base mandó la orden de monitorizar cada 30 segundos, se habrá transmitido tres veces el carácter 'a'. Y este tiempo de 90 segundos (tiempo del carácter $\times$ 3) será diferente para otro carácter. El motivo de enviar tres veces cada carácter es intentar reducir a cero el porcentaje de errores. Y es que en la implementación del canal nos hemos encontrado con un feroz enemigo: las colisiones.

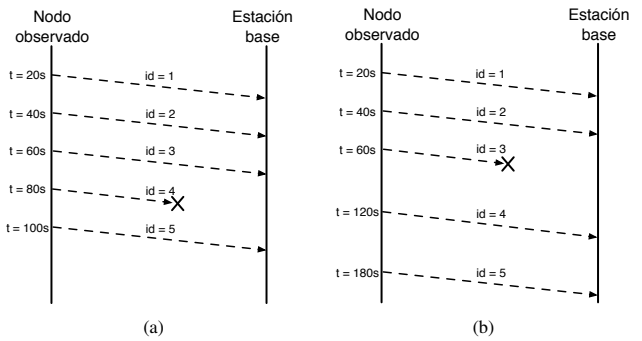


Figura 4. Diferentes circunstancias de colisiones

Debido a que el medio de transmisión inalámbrico es compartido, si un sensor recibe simultáneamente dos mensajes, ambos colisionan y por lo tanto ambos mensajes se vuelven incomprensibles. Esto puede llevar a confusiones por parte del sniffer a la hora de medir los tiempos transcurridos.

Las colisiones pueden afectar al nodo observado por Bob tanto durante el envío como durante la recepción de paquetes. Tras múltiples tests en nuestro escenario de pruebas (similar al de la Figura 3) se constata que las colisiones de los paquetes recibidos en este nodo suponen un 5 % del total de las colisiones. Sin embargo, las que más nos interesan, y que suponen un 95 % del total de las colisiones, son las que afectan a los paquetes que envía el nodo. A fin de detectarlas, los paquetes han de contar con un identificador en la cabecera para que el sniffer pueda compararlo con el identificador de paquetes consecutivos. Si el sniffer recibe el paquete con identificador 3 y a continuación recibe el paquete con identificador 5, calcula que entre ambas recepciones se perdió un paquete.

Las pérdidas de paquetes individuales pueden darse en dos circunstancias diferentes:

1. Se pierde un paquete durante un intervalo de tiempo en el que la tasa de envío se mantiene inalterada.
2. Se pierde el primer paquete que cambiaba el intervalo de tiempo respecto a su predecesor.

En la Figura 4 se observan las dos circunstancias posibles en un escenario en el que suponemos inicialmente $t = 20$ segundos y un cambio posterior a $t = 60$ segundos. En la primera se ve como el nodo espiado envía cuatro paquetes a la estación base, los tres primeros cada 20 segundos y un último paquete 40 segundos después del tercero. El nodo sniffer detecta la colisión del cuarto paquete, por lo que sabe que entre el instante 60 segundos y el instante 100 segundos

[2]La tendencia actual es incorporar células solares para alargar su vida útil (e.g. ECS310 de enocean®).

[3]En el mercado existen varias soluciones que permiten esta funcionalidad, como es el caso de Jackdaw [15]

se ha perdido un paquete. Tenemos que el tiempo transcurrido entre el envío del paquete tres y el envío del paquete cinco es 40 segundos. Si se supone, como es el caso, que el paquete que se ha perdido fue enviado transcurrido el mismo intervalo que el último recibido, sólo tenemos que coger los 40 segundos y dividirlos por 2, obteniendo que el intervalo transcurrido entre el paquete colisionado y el quinto paquete es de 20 segundos.

Para la segunda circunstancia, si utilizáramos la misma solución se obtendría que el tiempo transcurrido entre el paquete colisionado, el número tres, y el paquete cuatro es de 40 segundos, por lo que obtendríamos un tiempo erróneo. Esta circunstancia se soluciona almacenando en cada captura de tráfico el intervalo del último envío y restándoselo al tiempo transcurrido entre las dos últimas transmisiones con éxito. En este caso, con la captura del paquete número dos almacenaríamos el intervalo 20 segundos, al recibir el cuarto paquete transcurridos 80 segundos detectaríamos la colisión y a estos 80 segundos le restaríamos el intervalo almacenado anteriormente de 20 segundos, obteniendo el resultado de 60 segundos.

Aunque se sabe como solucionar ambos casos, el gran problema es que resulta imposible saber durante la ejecución en que circunstancia nos encontramos. Por ello, optando por alguna de las dos soluciones conseguiremos reducir el número de errores aunque no por completo. Además, este esquema podría requerir un pequeño cambio en el código de la aplicación de las motas de forma que se puedan numerar los paquetes (si el propio protocolo de envío de paquetes de la aplicación de las motas no lo hace por defecto). Si bien, el cambio es mínimo, como veremos en la sección V, de producirse, éste aumentaría la detectabilidad del canal por parte de agentes locales.

Dado que el requisito de integridad de los datos es esencial en nuestro escenario, en una segunda implementación del canal encubierto hemos debido corregir las colisiones mediante la replica de intervalos (y el control de estas repeticiones para mantener la integridad del mensaje decodificado), reduciendo así, desafortunadamente, el ancho de banda de manera drástica. No obstante, como habiamos extraido en III-B, se trata del requisito menos estricto de todos.

Tras multiples simulaciones, en esta segunda implementación hemos logrado un ancho de banda de $1/230$ caracter/seg. Esto significa que para comunicar en nuestro escenario el identificador de referencia de un contenedor de 10 caracteres, por ejemplo; Alice necesitaría unos 38 minutos[4]. La tasa de error detectada inherente al escenario y el funcionamiento que realiza es del 4 %.

V. Estudio de detectabilidad

Debido a la naturaleza inalámbrica de las redes de sensores, un atacante podría escuchar las transmisiones e incluso inyectar tráfico en la red; especialmente si la red está desplegada en un entorno hostil. Por ello, la seguridad en redes de sensores se centra en proteger cuatro aspectos: confidencialidad, integridad, disponibilidad y la vida de la batería. Dado que nuestro canal no afecta a ninguno de estos servicios, eludiría la mayoría de las soluciones de seguridad actuales.

Como ya se ha comentado con anterioridad, las redes de sensores son un tipo de red ad-hoc inalámbricas cuyas diferencias hacen inviable la aplicación de IDSs (*Intrusion Detection Systems*) desarrollados para redes ad-hoc. Para empezar, la capacidad de los nodos impide instalar un agente de detección completo. Por ello, se usan soluciones parciales como:

- Analizar las fluctuaciones en las lecturas de los sensores
- Analizar la integridad del código
- Vigilar la información intercambiada entre los sensores

Ante un IDS dedicado al análisis de las lecturas de los sensores, nuestro canal encubierto pasaría totalmente desapercibido puesto que no altera dichos valores. De igual forma, la vigilancia de la información intercambiada entre los sensores no pondría de manifiesto el canal encubierto puesto que no se modifica ninguno de los campos de los paquetes. A su vez, un exhaustivo estudio del código que forma el programa de las motas tampoco supondría un problema, ya que la única funcionalidad sospechosa (los envíos de cambios de tiempo de monitorización) se realizarían desde la estación base y sin necesidad de modificar su código.

En [17] se propone una solución de IDS específico para redes de sensores. Esta solución considera dos tipos de agentes: locales y globales. Los agentes locales monitorizan tanto las operaciones realizadas como la información enviada y recibida por el nodo. Por tanto, los agentes locales detectarían los ataques que afecten la integridad física o lógica de la mota así como el intento de influenciar en la recogida de datos por parte de entidades no autorizadas.

Por otro lado, los agentes globales vigilan las interacciones con sus vecinos inmediatos, comportándose a modo de guardián que analiza y procesa el contenido paquetes. Estos agentes serían capaces de detectar si un nodo está borrando o modificando algún campo de los paquetes intercambiados por las motas antes de retransmitirlo. En el caso de detectar alguna amenaza de seguridad, el agente generaría información de alerta y la enviaría a la estación base.

Dado que nuestro canal no modifica el estado de los nodos ni la información contenida en los paquetes de manera arbitraria, los agentes descritos no serían capaces de detectarlo. Sí podrían levantar ciertas sospechas los continuos mensajes de actualización de la tasa de transferencia por parte de la estación base. Aunque la estación base es un nodo autorizado, y, además, su comportamiento entraría dentro del uso normal de la red. No obstante, continuos cambios en estos tiempos (si queremos enviar un mensaje oculto con 10 caracteres, se cambiarían los tiempos de monitorización 20 veces en un intervalo de tiempo relativamente corto, ya que utilizamos también un carácter de *cambio* por cada cambio de caracter), si el mensaje enviado en el canal encubierto es muy largo, podrían levantar sospechas en el caso de que se produzca un análisis detallado de esta frecuencia.

Una posible solución ante este problema sería utilizar varias motas para enviar los datos ocultos. De esta forma, cada uno

[4]La adecuación del tiempo necesario para transmitir un mensaje será relativo al objetivo que se persigue en la comunicación oculta.

de los nodos no recibiría un volumen notorio de notificaciones de cambio de intervalo, y los agentes locales de estos nodos no verían como una situación anómala el recibir un número dado de notificaciones, que se vería reducido en función del número de motas utilizadas.

Para que esta solución tuviese sentido es necesario que el sniffer pueda monitorizar varios nodos de manera simultánea, ya sea porque la ganancia de su antena se lo permite o porque tiene varias antenas en distintas ubicaciones. Asimismo, sería necesario establecer una secuencia predefinida de motas a observar por parte de Bob, por lo que la implementación y sincronización del canal se hace más compleja. Nótese que en este último caso se abren nuevas vías de ocultación. Se podría investigar, por ejemplo, desarrollar un canal encubierto de conteo (i.e., se codifican los caracteres en función del número de nodos que envíen en un intervalo) o bien un canal de ordenación (i.e., el orden de envío indica la información).

El empleo de técnicas de detección dependerá siempre de las características de la aplicación y del escenario concreto sobre el que dicha aplicación se ejecuta. Dada la sobrecarga que pueden introducir estos mecanismos, en términos de transmisión sobre el medio inalámbrico y de procesamiento y almacenamiento en los nodos, su uso puede resultar justificable únicamente en aplicaciones con fuertes requisitos de seguridad y en las que los dispositivos involucrados dispongan de la suficiente capacidad y autonomía como para que la ejecución de un sistema de detección no imponga una limitación intolerable sobre las prestaciones ofrecidas al usuario final.

Así pues, se puede concluir que nuestro canal de comunicación oculto es lo suficientemente difícil de detectar como para pasar desapercibido ante sistemas de detección usuales. No obstante, agentes locales que llevaran a cabo un análisis en la estadística de los cambios de frecuencia en los mensajes de monotorización de las motas podrían elevar las alertas necesarias para comenzar a investigar la existencia de dicho canal.

VI. Conclusiones

En este artículo hemos diseñado una canal encubierto sobre una red de sensores con un ancho de banda muy limitado pero, hasta donde alcanza nuestro conocimiento, es el primer intento de análisis, diseño e implementación de este tipo de canales en este entorno.

Para ello hemos ideado un escenario ficticio y extraído los requisitos principales del canal. A partir de estos se ha desarrollado un canal de temporización multitasa y se ha llevado a cabo un estudio sobre la detectabilidad de este tipo de comunicaciones en redes de sensores. Si bien el ancho de banda del canal es mejorable, hemos preferido, acorde con los requisitos extraídos, primar la integridad de los datos enviados.

Asimismo hemos encontrado nuevas vías de ocultación que estamos analizando en la actualidad. En concreto, se tratan de canales de almacenamiento y modificación del enrutamiento; por lo que habrá que analizar detenidamente sus implicaciones a nivel de detectabilidad y de funcionamiento de la red. Más específicamente, tres son los candidatos (que presentan sus desventajas y ventajas asociadas): canal encubierto sobre uIP, utilización de los campos de protocolos de enrutamiento (RSSI y LQI), y modificación de las rutas seguidas por los paquetes.

También resulta de interés indicar que se ha llevado a cabo la implementación del canal propuesto como prueba de concepto sobre el simulador de Contiki y evaluado su funcionamiento desplegándolo sobre un número muy reducido de motas físicas. Se hace por tanto necesario llevar a cabo un despliegue sobre una red de sensores real con objeto de hacer mediciones más fiables.

Agradecimientos

Este trabajo ha sido sido financiado parcialmente por el Ministerio de Ciencia e Innovación y de la Junta de Andalucía a través de los proyectos ARES (CSD2007-00004) y FISICCO (P11-TIC-07223), respectivamente.

Referencias

[1] B. W. Lampson, "A Note on the Confinement Problem," *Commun. ACM*, vol. 16, no. 10, pp. 613–615, 1973.

[2] S. L. Brand, "Department of Defense Trusted Computer System Evaluation Criteria - The Orange Book," U.S. Department of Defense, Tech. Rep. DoD 5200.28-STD, 1985. [Online]. Available: http://csrc.nist.gov/publications/history/dod85.pdf

[3] Center for Advanced Internet Architectures, "Covert Channels in Computer Network Protocols Bibliography," March 2014. [Online]. Available: http://caia.swin.edu.au/cv/szander/cc/cc-cnetworks-bib.html

[4] S. Li and A. Ephremides, "Covert channels in ad-hoc wireless networks," *Ad Hoc Networks*, vol. 8, no. 2, pp. 135 – 147, 2010.

[5] G. J. Simmons, "The Prisoners' Problem and the Subliminal Channel," in *Advances in Cryptology: Proceedings of CRYPTO*, ser. LNCS, D. Chaum, Ed. Santa Barbara, California, USA: Plenum Press, August 21-24 1983, pp. 51–67.

[6] C. Meadows and I. S. Moskowitz, "Covert Channels – A Context-Based View," in *Proceedings of the First International Workshop on Information Hiding*. London, UK: Springer-Verlag, 1996, pp. 73–93.

[7] J. Shen, S. Qing, Q. Shen, and L. Li, "Optimization of Covert Channel Identification," in *SISW '05: Proceedings of the Third IEEE International Security in Storage Workshop*. Los Alamitos, CA, USA: IEEE Computer Society, 2005, pp. 95–108.

[8] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292 – 2330, 2008.

[9] J. Al-Karaki and A. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, no. 6, pp. 6 – 28, dec. 2004.

[10] L. Krishnamurthy, R. Adler, P. Buonadonna, J. Chhabra, M. Flanigan, N. Kushalnagar, L. Nachman, and M. Yarvis, "Design and deployment of industrial sensor networks: Experiences from a semiconductor plant and the north sea," in *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, ser. SenSys '05. New York, NY, USA: ACM, 2005, pp. 64–75. [Online]. Available: http://doi.acm.org/10.1145/1098918.1098926

[11] "Contiki: The open source os for the internet of things." [Online]. Available: http://www.contiki-os.org/

[12] "Tinyos official website." [Online]. Available: http://www.tinyos.net/

[13] S. C. Cabuk S., Brodley C.E., "Ip covert timing channels: Design and detection," in *Proceedings of the 11th ACM Conference on Computer and Communications Security*. ACM Press, 2004.

[14] S. Cabuk, C. E. Brodley, and C. Shields, "Ip covert channel detection," *ACM Trans. Inf. Syst. Secur.*, vol. 12, pp. 22:1–22:29, April 2009.

[15] "Rzraven usb stick (jackdaw)." [Online]. Available: http://www.ibr.cs.tu-bs.de/projects/mudtn/doxygen/a01892.html

[16] D. Huffman, "A method for the construction of minimum-redundancy codes," in *Proceedings of the I.R.E.*, editor, Ed., pp. 1098–1102.

[17] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *IEEE Consumer Communications & Networking Conference (CCNC 2006)*, IEEE. Las Vegas (USA): IEEE, January 2006, pp. 640–644.

Índice de autores

ISBN: 978-84-9717-323-0